

Distributed Media Server Architecture for SIP using IP Anycast

Ladislav Anel
Tekelec

Prague, Czech Republic
ladislav.anel@tekelec.com

Jiri Kuthan
Tekelec

Berlin, Germany
jiri.kuthan@tekelec.com

Dorgham Sisalem
Tekelec

Berlin, Germany
dorgham.sisalem@tekelec.com

Abstract—To support a wide variety of service scenarios including NAT traversal, lawful interception or transcoding, a VoIP provider has to not only use signaling servers but also media servers. To reduce the load on the provider in terms of bandwidth usage and reduce the effects of adding media processing servers in the media path, the media servers should be located in the proximity of the users. In this paper we investigate the possibility of using IP anycast for enabling users to detect and use media servers in their proximity. We propose four different approaches and give a high level evaluation of the pros and cons of each approach.

I. INTRODUCTION AND MOTIVATION

The session initiation protocol (SIP) [1] is increasingly becoming the de-facto standard for VoIP deployments in fixed and wireless networks. In theory, a VoIP provider needs only to consider SIP signaling messages and does not have to deal with the exchanged audio and video media data. As illustrated in Fig. 1 a provider offering SIP-based VoIP services requires only SIP components that process SIP messages and act as a rendezvous point between the caller and callee. A caller wishing to make a call sends his session initiation requests to these components which forward the requests to the callee. The result of the session initiation would be that the caller and callee know the IP addresses of each other and can exchange the media streams among each other directly without the involvement of the SIP components of the provider.

However, in reality VoIP providers have to deal with the exchanged multimedia content as well. This can be due to various reasons:

- Lawful interception: In various countries, VoIP providers are required by law to be able to provide law enforcement agencies with the signaling and media traffic sent or received by a certain user. This requires the provider to not only process signaling

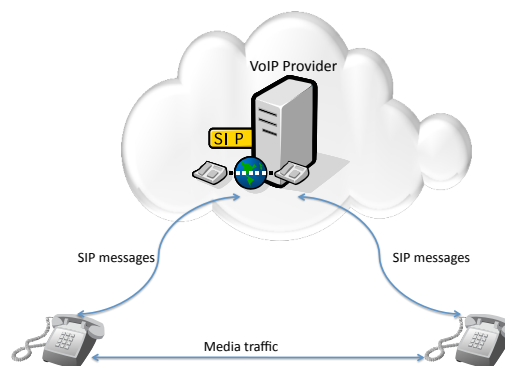


Fig. 1. Centralized VoIP infrastructure

messages but also the media content exchanged by all of its subscribers.

- Security and traffic policy: To hide internal SIP servers, application servers and other components from the end users and to monitor and police the traffic sent by the users, providers often deploy session border controllers (SBC). SBCs act as the interface between the provider and the users and have the task of filtering malicious traffic, ensuring that the users do not consume more bandwidth than they are allowed to and hide the provider's SIP infrastructure from end users. To be able to control and police the users' behavior, the SBCs must also process the users' media traffic.
- NAT traversal: Using SIP end users can exchange information about their whereabouts, i.e. the IP addresses they are currently using. These IP addresses can then be used to exchange signaling and media traffic between the users. In case a user is behind a NAT and is using a private IP address then direct communication between the users is not possible. Depending on the type of the NAT some NAT

traversal solutions for SIP such as TURN [2] or RTP relay [3] introduce a media handling component between the users that accept traffic from one user behind a NAT and send it to another user behind another NAT.

- **Transcoding:** In some cases the caller and callee can not use the same audio or video compression style or do not share the same capabilities, i.e., one side can only accept speech and the other side can only accept text. In such scenarios it is necessary to include a media transcoding entity in the media path that adjusts the characteristics of the exchanged traffic in accordance with the capabilities of the users.

In the context of this paper we will call the components responsible for providing the media processing capabilities media servers. A simple approach for introducing media servers to a VoIP infrastructure is by placing the servers in a central location, similar to the approach illustrated in Fig. 2. This means, however, that all media traffic sent or received by the provider’s users would traverse the provider’s infrastructure and would increase the costs for the used bandwidth. Having to traverse a central location before getting forwarded to the final destination could also have negative impact on the media traffic itself as it can considerably increase the transmission delay. A

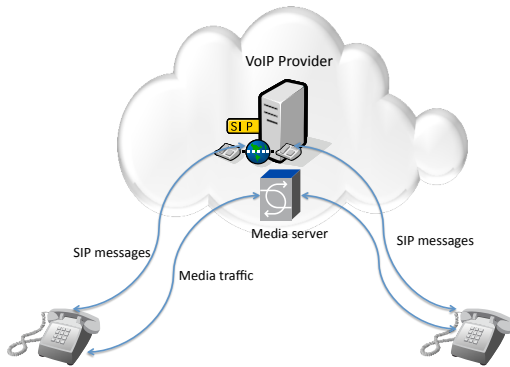


Fig. 2. Centralized VoIP infrastructure

more optimal approach is to separate the media servers from the signaling servers and to distribute the media servers and locate them closer to the users, see Fig. 3. This decreases the bandwidth costs for the provider and reduces the effects the possible negative effects on the traffic in terms of increased delay.

The challenge of a distributed media server architecture is to enable the user devices to discover a close-by media server to use without having to extend the logic of

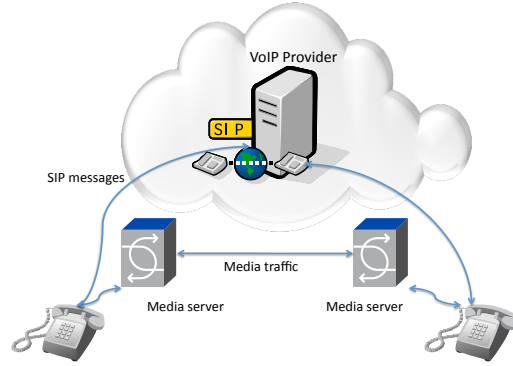


Fig. 3. Distributed media servers in a VoIP infrastructure

the devices, using new protocols or having to configure the user devices.

In this paper we use IP anycast technology [4] for routing media traffic through media servers that are located in the proximity of the users. We propose different architectural solutions for using anycast and compare the advantages and disadvantages of each approach.

In Sec. II we provide a short introduction to SIP and anycast. The resilience of IP anycast towards changes in the routing tables are measured in Sec.III. Some of the restrictions that need to be considered when introducing media handling components in a VoIP network are presented in Sec. IV. In Sec. V four approaches for using anycast for enabling users to use media handling components in their proximity are provided. These approaches are evaluated in Sec. VI. In Sec. VII a summary of the work and a look at the further steps needed to be considered are presented.

II. BACKGROUND AND RELATED WORK

A. The Session Initiation Protocol

The most important SIP [1] operation is that of forwarding SIP requests between subscribers. To achieve this functionality we can distinguish different SIP entities:

- **Proxy:** The proxy provides the routing logic of the VoIP service. When a proxy receives a SIP request from a user agent or another proxy it also conducts service specific logic, such as checking the user’s profile and whether the user is allowed to use the requested services. The proxy then either forwards the request to another proxy or to another user agent or rejects the request by sending a negative response.

- **Redirect:** A redirect server receives a request and informs the caller about the next hop server. The caller then contacts the next hop server directly.
- **User Agent:** A logical entity in the terminal equipment that is responsible for generating and terminating SIP requests. The UA can be the VoIP application used by the user, e.g., the VoIP phone or software application, a VoIP gateway which enables VoIP users to communicate with users in the public switched network (PSTN) or an application server, e.g., multi-party conferencing server or a voicemail server.
- **Registrar:** To assist SIP entities in locating the requested communication partners SIP supports a further server type called registrar server. The registrar server is mainly thought to be a database containing locations as well as user preferences as indicated by the user agents.

In SIP, a user is identified through a SIP URI in the form of `user@domain`. This address can be resolved to a SIP proxy that is responsible for the users domain. To identify the actual location of the user in terms of an IP address, the user needs to register his IP address at the SIP registrar responsible for his domain. Thereby when inviting a callee, the caller sends his invitation to the SIP proxy responsible for the callee’s domain, which checks in the registrars database the location of the callee and forwards the invitation to the callee. The callee can either accept or reject the invitation. The session initiation is then finalized by having the caller acknowledging the reception of the callees answer. During this message exchange, the caller and callee exchange the addresses at which they would like to receive the media and what kind of media they can accept. After finishing the session establishment, the end systems can exchange data directly without the involvement of the SIP proxy. For authenticating a user SIP uses the digest authentication mechanisms, which is based on a challenge/reply approach.

B. IP Anycast

IP Anycast is a network technique which allows a client to access the nearest host of a group of hosts that share the same anycast IP address, where the nearest host is defined according to the routing system’s measure of distance. It is also referred to as one-to-any communication where “any” means one host of the anycast group. Usually, the hosts in the anycast group are replicas, able to provide the same service. To take an advantage of anycast, servers are distributed topologically and geographically across the Internet. An

IP anycast deployment solely depends on the network, routers and routing protocols.

The scale of anycast deployment within the routing system can vary from a small local network up to a large scale network spreading over the global Internet [5]. Figure 4 shows the basic idea of a network-layer (IP) anycast deployment.

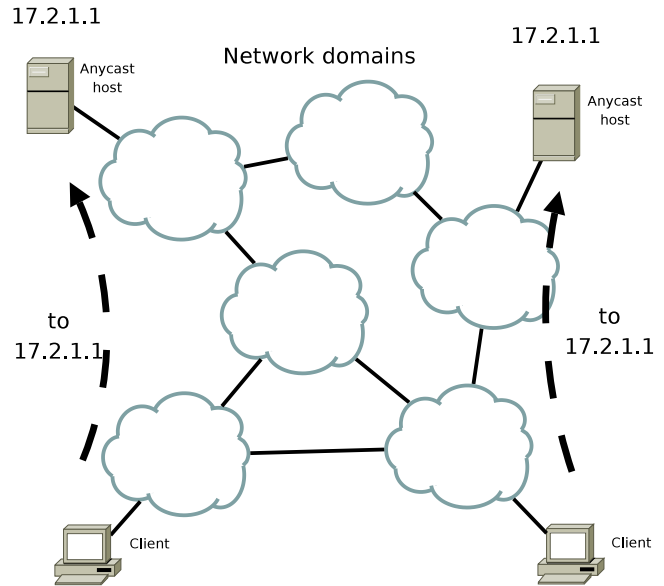


Fig. 4. IP anycast mechanism

Patridge et al. [4] originally proposed the idea of anycast at the network-layer (IP). They defined that anycast is a stateless best effort delivery of an anycast datagram to at least one host, and preferably only one host. In RFC4786 [6] Abley and Lindqvist cover the best current practices of using IP anycast.

Ballani et al. [5] state that deployment of IP anycast is currently limited to just query/reply services such as for DNS root servers [7], primarily to spread the load as a defence against DoS attacks.

1) *UDP, TCP transports and Anycast:* It is important to remember that routing in the Internet is stateless. This means that an anycast network has no obligation to deliver two successive packets sent to the same anycast address to the same host. This might happen when a client is topologically in the middle of two anycast hosts with equal-cost paths.

a) *UDP:* Since UDP transport is connectionless and anycasting is a stateless service, UDP can treat anycast addresses like regular IP addresses. A UDP datagram sent to an anycast address is just like a unicast UDP datagram from the perspective of UDP and its application.

Any cast is best suited for services with short and simple transactions that can be realized in a query-reply

kind of interactions such as DNS services.

Some services have long transaction times and need to exchange more datagram between the client and the anycast host. In this case a change in the routing tables might cause two UDP messages belonging to the same transaction to be delivered to two different hosts.

b) TCP: TCP’s use of anycasting is less straightforward because TCP is stateful. It is hard to envision how one would maintain TCP state with an anycast server when two successive TCP segments sent to the anycast server might be delivered to completely different hosts.

Engel et al. [8] propose a solution for this problem. This proposal is based on minor modifications of the TCP/IP stack at the host part where the anycast service is running. It does not require any modifications to routers and routing protocols. These modifications are limited to changes at the IP layer of the recipient of the TCP connection, making this scheme suitable to a client/server environment [4]. The basic idea is to pin the end-host to which the first packet of the flow has been sent. The pinning is done by inserting a loose source route option in all subsequent packets of the same TCP flow.

2) *Pros and Cons of Anycast:* General pros and cons of anycast technology include:

Pros:

- locality/latency improvements by reducing network distance between clients and servers (at least eliminating the worst case)
- high availability - provides a service without outages
- reduce list of addresses of geographically dispersed servers to a single distributed anycast address

Cons:

- IP anycast wastes the address space (the longest IP prefix is /24), even though only one IP address is used for running the service. This stems from the fact that routing protocols such as BGP [9] do not propagate single IP addresses so as to keep routing tables to manageable sizes.
- IP anycast does not always offer the nearest anycast server (latency-based proximity).
- Changes in routing tables propagate only slowly. This means that after the failure of a host that was part of the anycast group, it will take some time till the routing tables are updated. Till the tables are updated, traffic might still be routed to the failed host. This makes the service to be partially unavailable for some time duration ranging from a few seconds to many minutes [10]).
- As the routing might change during the duration of a session, IP anycast is generally considered as

not suitable for “long lived” sessions such as TCP connections. Ballani et. al. [11] investigated IP anycast with regard to proximity¹ and affinity² and state that IP anycast is also a good candidate for using services based on TCP or applications with long-lived sessions. Measuring the performance of IP anycast in a global deployment the authors indicate that 93.75% of the source-destination pairs never changed (probability of selecting the same anycast node). In other words, the probability that a two minute (or one hour) connection would experience a change is roughly 1 in 13000 (or 1 in 450).

III. EVALUATION OF IP ANYCAST

In this section we present the results of real-live measurements that we conducted with goal of evaluating how fast the failure of a host is detected and the routing tables are updated so that traffic is no longer routed to the failed server. The measurements were conducted over the PlanetLab³. We setup an anycast address with two nodes located in Berlin and Prague.

From 142 nodes in the PlanetLab testbed ICMP echo requests [12] were sent to the anycast address. On each of the used PlanetLab hosts two scripts were installed. One of them measured ICMP echo replies in 8 seconds intervals and the other one was collecting results from traceroute in 2 minutes intervals to see what anycast node is used for a particular PlanetLab host.

As indicated in Table I most of planet-lab hosts directed ICMP packets to the Prague node. It shows that from a routing path prospective Prague anycast node is situated within an ISP (Internet Service Provider) on a back-bone that is more accesible in terms of routing metrics from the Internet.

Anycast node	No. of hosts routed to the node
Prague	103
Berlin	39
Σ	142

TABLE I
DESTINATIONS OF PLANET-LAB HOST ICMP REQUESTS

Figure 5 depicts the convergence times of ICMP packets originally destined to Prague and then re-routed to Berlin after taking the Prague node off the network.

¹ability to find close-by members of the anycast group.

²tendency of subsequent packets of a “connection” to be delivered to the same target.

³www.planetlab.org

We can observe that most of the nodes manages to shift their ICMP requests to the Berlin host in less than 20 seconds. Note that SIP uses a timeout to discover the failure of a host. This timer is generally set to maximally 32 seconds. This means that if a caller initiated a call just after the failure of a destination, the initial request and the first couple of retransmissions would get lost but the IP anycast infrastructure would converge fast enough for the call initiation to succeed before the SIP timeout expires.

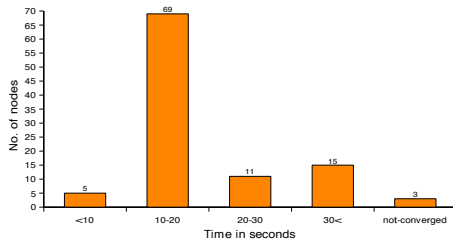


Fig. 5. Route convergence time to Berlin's anycast node

Figure 6 depicts the convergence times of ICMP packets originally destined to Berlin and then re-routed to Prague after taking the Berlin node off the network. We can observe that most of the nodes manages to shift their ICMP requests to the Prague host in less than 10 seconds.

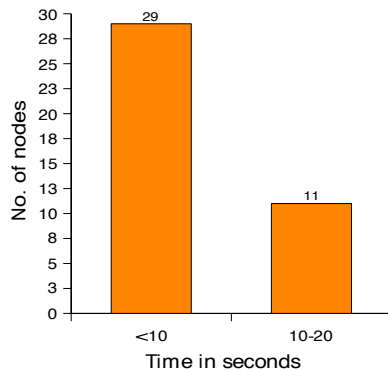


Fig. 6. Route convergence time to Prague's anycast node

The measurements suggest that IP anycast has in general an acceptable performance for rerouting traffic in case a host fails. Actually only 3 planet-lab hosts could not reach a new destination. Interestingly, all 3 planet-lab hosts are situated in Italy.

IV. USING AND DISCOVERING MEDIA SERVERS

As mentioned before, providers of VoIP services will in some scenarios need to deal not only with signaling

traffic but also with audio or video traffic. To reduce the costs of bandwidth and optimize the transmission delay of the media data, the media servers will have to be deployed in a distributed manner and in the proximity of the users. As a example of a media server we used a component called RTP proxy. RTP proxies are a special kind of media servers that are used for supporting NAT traversal. When a SIP proxy receives a SIP request and identifies that an RTP proxy is needed, it signals the RTP proxy that all traffic coming from a certain address should be forward to a certain address. The SIP proxy also manipulates the SDP content of the SIP messages so that the caller and callee send their media traffic to the RTP proxy and are willing to accept media traffic from the RTP proxy.

When developing an architecture for distributing media servers one needs to accommodate the following restrictions:

- **Ease of deployment:** The configuration overhead for adding or removing a media server should be low and with no need to update or change user agents.
- **Interoperability:** The used technology should not require the user agents to deploy new protocols or extensions to the SIP specifications [1]
- **Proximity:** The choice of which media server should be used to serve a certain user should take the user's location in consideration and choose a server that is in the proximity of the user so as to reduce the transport delay of the media traffic
- **Resilience:** The failure of a media server could have some negative effects on calls served by this server. The used technology should aim at reducing these negative effects by detecting the failure and possibly rerouting the media traffic to another media server. Of more importance though is that the failure of a server does not lead to the rejection of new calls. Further, changes in the IP routing tables should not cause an interruption to the service.

Media servers are already widely used today in various VoIP deployments. To enable a user agent to discover and use a media server different approaches are currently used:

- **Static configuration:** User agents are configured with the address of a SIP server that is close to them and which they are supposed to contact for registering and as a first contact point when initiating a call. Attached to the SIP server is the media server. After receiving a SIP request, the SIP server instructs the user agent to forward the media traffic to the attached media server. This approach allows for a distributed architecture as different

user agents can be configured with the addresses of different SIP servers. Further, the addresses of the SIP servers are usually presented as names that are resolvable using DNS and can be mapped to different servers. Once the server with whom a user agent is communicating fails, the user agent can contact another server as resolved by DNS. However, the provider will have to configure the user agents to use certain servers. Changes in the naming structure, or distribution of users to servers requires changes to the configuration which makes the approach impractical.

- DNS: Instead of manually allocating users to media servers, all user agents can use the same address of the VoIP provider. This address is resolved using DNS and the DNS server resolves the address of the VoIP provider differently to different users. This could be achieved by having different SIP servers and using round-robin strategy for replying to different requests. To achieve geographic proximity the DNS logic might be extended with sufficient information about the mapping of IP addresses to geographic locations. When receiving a request for resolving the name of a VoIP server, the DNS server uses this information to choose a SIP server that is in the proximity of the requesting user agent. This approach is often used with content distribution networks [13]. This requires, however, maintaining an up-to-date mapping between IP addresses and geographical locations as well as a complex DNS server that can take this information into account.
- DHCP: Using the Dynamic Host Configuration Protocol (DHCP) [14], [15] a host can discover and contact a DHCP server. The DHCP server provides the host with an IP address as well as the addresses of a DNS server and a default router that can be used for routing traffic to the Internet. DHCP is extended for SIP [16], [17] so that DHCP servers also inform the SIP user agents about a SIP server to be used. Similar to the previous approaches, the media server is attached to the SIP server.
- TURN: Traversal Using Relay NAT (TURN) enables a user agent behind a NAT to discover the address of a media server that has a public IP address. This media server acts as the public interface for the user agent and relays all media traffic sent by the user agent with its own public IP address. It further receives media traffic destined to the user agent and relays it to it. This approach, however, is limited to the NAT traversal scenario, requires the user agents to implement the TURN protocol and does not take proximity into account.

V. USING IP ANYCAST FOR DISCOVERING MEDIA SERVERS

IP anycast technology provides the means for detecting servers in the proximity of a client. Further, a failed server stops advertising its availability and is eventually removed from the routing tables. Client requests would, hence, no longer be forwarded to the failed servers. That is, IP anycast already fulfills a good deal of the restrictions on technologies for deploying distributed media servers as presented in IV.

In the following we present different concepts of how to deploy IP anycast for the discovery of media servers without having to update or extend SIP user agents. The proposals presented here are specially targeting the usage of RTP proxies.

The simplest approach for using IP anycast for distributing RTP proxies would be to send SIP requests to a SIP proxy which would discover that the media traffic has to traverse a RTP proxy. The SIP proxy would then include the IP anycast address of the distributed RTP proxies in the SDP part of the SIP message. The end users can then send their traffic to the announced address of the RTP proxies. However, Before an RTP proxy can start forwarding RTP packets between two users it has to get the information that RTP packets coming from a certain address should be forwarded to a certain address. This information is only known to the SIP proxy which is dealing with the processing of SIP messages. The SIP proxy that is dealing with NAT traversal needs hence to know which RTP proxy is closest to its subscribers. Therefore, in the following scenarios some SIP logic is located close to each RTP proxy.

A. Anycasting Geographically Spread DNS Servers

With the DNS-based method, the VoIP provider's DNS server that is responsible for resolving the name of the VoIP provider into an IP address consists of geographically dispersed servers listening to an anycast address. These DNS servers are co-located with the media servers and SIP servers which use unicast addresses.

A user wishing to send a request to a subscriber of the VoIP provider starts by resolving the VoIP provider's address using DNS. The DNS query is sent to the anycast address of the provider's DNS servers and is directed to the one closest to the user. The found DNS server returns the unicast address of the SIP proxy co-located with the DNS server. Afterwards the SIP client exchanges traffic with the discovered SIP and media server and the anycast IP address does not play any other role until DNS relookup. Figure 7 displays the interaction between the different components and the exchanged messages.

The discovered SIP server processes the SIP requests of the user and forwards them to the provider's VoIP servers which provide the actual VoIP services such as authentication and advanced services. The media server co-located with the SIP server provides the media related processing services.

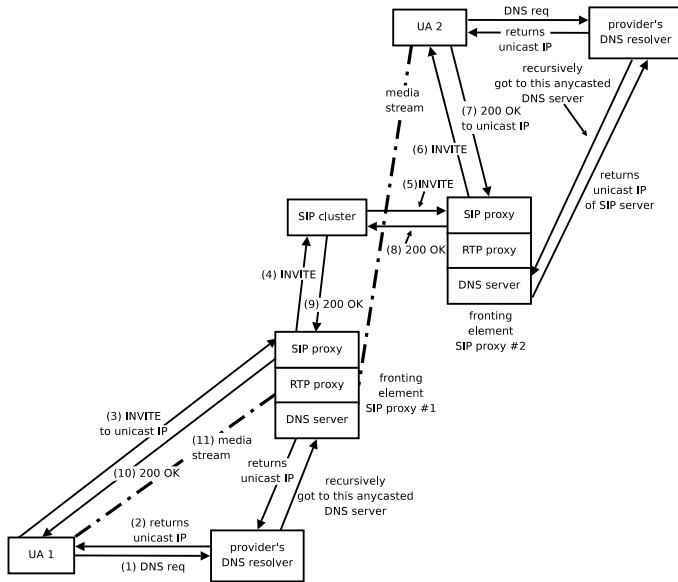


Fig. 7. Anycasting DNS servers

B. Anycasting SIP Servers

In this alternative, SIP servers are also located with the media servers and listen to an anycast for discovery purposes. The discovered SIP server acts as a proxy of the requests to the provider's VoIP servers which provide the actual VoIP services such as authentication and advanced services. The media server co-located with the SIP server provides the media related processing services.

Due to routing instabilities, requests sent during the same session might end-up at different SIP servers. Therefore, the SIP server should not maintain any state information about the running calls as otherwise SIP requests arriving at a different server can not be processed. This is, however, not entirely possible. On the transport layer, use of TCP breaks this requirement. On the SIP layer, the proxy can be stateless. In any case, all the anycast SIP servers must use the same rules for producing the transaction ids (branch Via parameter) as otherwise down-stream SIP servers will not match requests belonging to the same transactions during routing instabilities. Figure 8 displays the interaction between the different components and the exchanged messages. Figure 9 displays the message flow in this scenario.

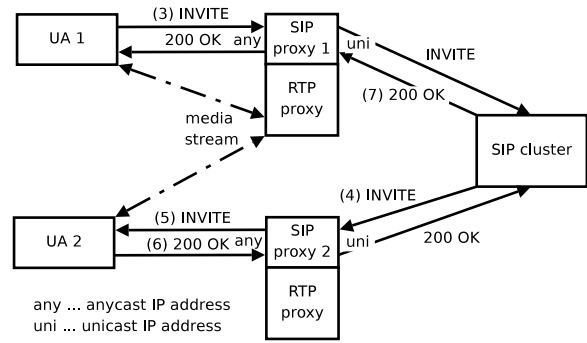


Fig. 8. Anycasting SIP servers scenario

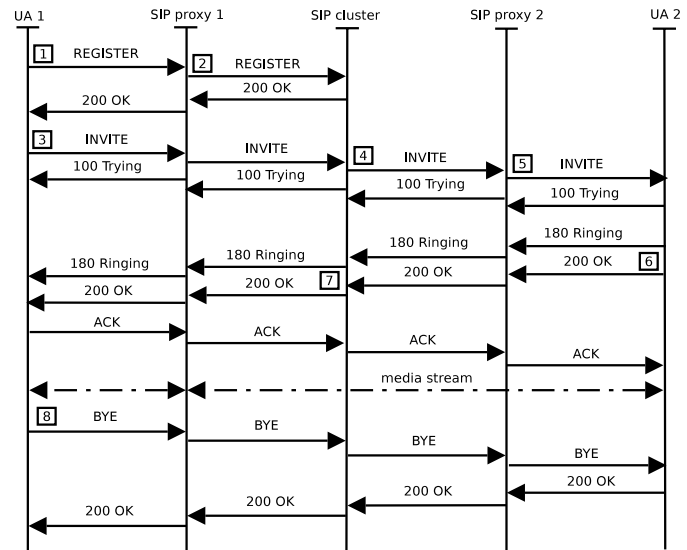


Fig. 9. Anycasting SIP server call flow

1) *TCP Persistent Connection Issue:* The key problematic part of anycasting SIP proxy servers is routing instability issues. In case a SIP client uses TCP transport for sending SIP messages it needs to create a TCP connection with the SIP server. The TCP connection must, however, be kept persistent as in case a re-routing occurs the connection is lost and TCP ACK might reach another SIP server which has no knowledge of the transaction and the TCP connection, see Figure 10.

C. Anycasting SIP Tunnels

In the previous approaches the media servers were co-located with a SIP component. The SIP component was responsible for receiving and processing the SIP messages sent by the users and forwarding them to other SIP servers. Further, these SIP components were responsible for controlling the media servers, i.e., opening a port for NAT traversal or instructing the media server to transcode incoming media in a certain way.

With the anycasting of SIP tunnels, the nodes hosting the media servers are connected to the SIP server of the

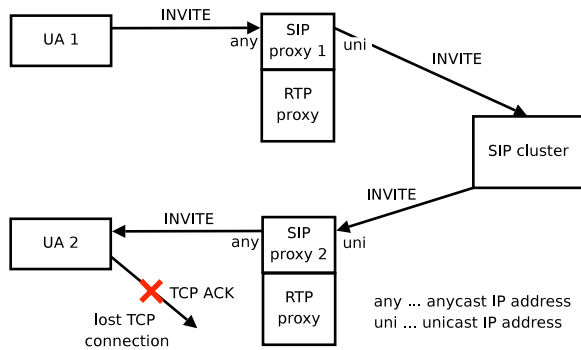


Fig. 10. Broken TCP persistency

provider over IP tunnels. A user wishing to send a SIP message sends it to an anycast address. This address is the address of the SIP server as indicated in DNS. The nodes hosting the media servers listen to this anycast IP address and when receiving requests over this address forward them to the SIP server. The media servers also include in the forwarded request information about their location. Forwarding the SIP messages over IP tunnels produces IP packets as if they came directly from a UA. The SIP servers can now instruct the users to use the media server located on the node indicated in the received packets.

An issue with this type of distribution is that the SIP servers need to remotely control the media servers. As all the SIP logic is located only at the SIP servers, the SIP servers will have to instruct the media servers about which ports to open and which IP addresses to connect. Depending on the distance between the media servers and the SIP servers this can add additional delay to the session establishment.

Figure 11 displays the interaction between the different components and the exchanged messages.

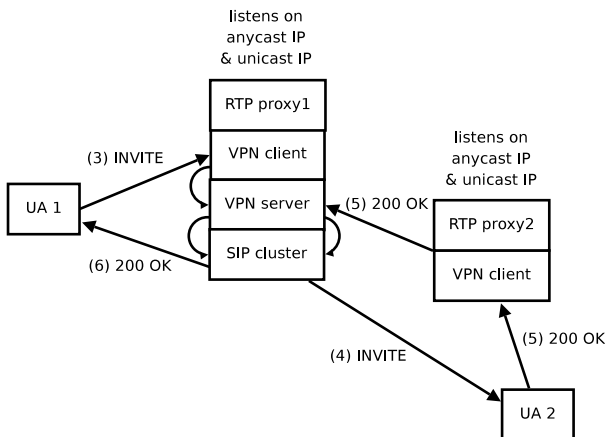


Fig. 11. Anycasting SIP tunnels

1) *Call Flows:* In the following the call sequence when using an RTP proxy for NAT traversal is explained, see Figure 12

a) REGISTER:

1. UA1 sends REGISTER to anycast IP address that gets forwarded to the SIP server via an IP tunnel.
2. SIP server replies with 200 OK directly to UA1.

b) INVITE:

3. UA1 sends an INVITE request to the IP anycast address of the VoIP provider. The request reaches a node listening to the anycast address and gets forwarded over an IP tunnel to the SIP server. The tunneled packets are marked with the identity of the media server running on that node. At the end of the tunnel the INVITE gets de-capsulated and delivered to the SIP server listening also at the anycast IP address.
4. The SIP server record-routes the INVITE and sends it directly to UA2.
5. UA2 replies with 200 OK. The message can go through different anycast tunnels but always gets delivered to the same SIP server where the call was initiated.
6. The SIP server matches the transaction and replaces the IP addresses in SDP body with the unicast IP address of media server #1 as indicated at the beginning of transaction and sends to UA1 directly.

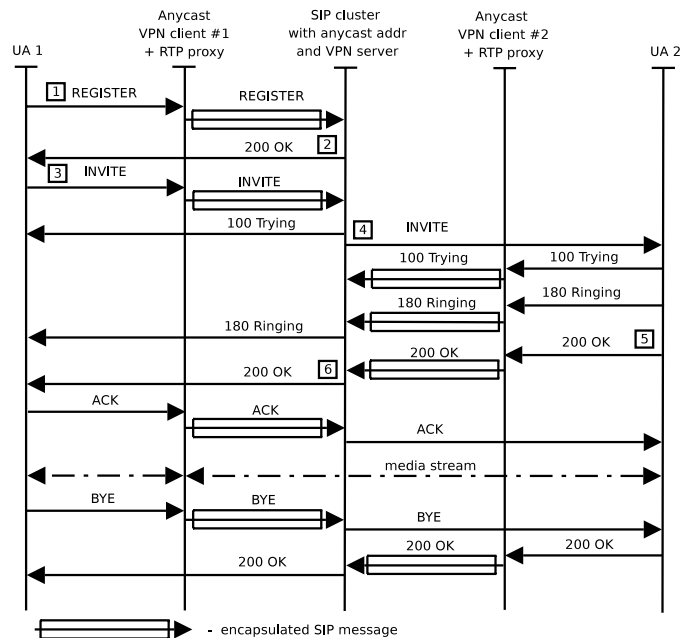


Fig. 12. Anycasting SIP tunnels - call flow

D. Anycast “bootstrap” Redirect Service

This concept is based on selecting a media server using SIP redirection. User agents send their initial INVITE requests to the main SIP server of the VoIP provider. The SIP redirects the INVITE requests to an anycast SIP server with a co-located media server. This server redirects the INVITE back to the main server after adding location information in the reply. The main SIP server uses the location information to steer the proper media server and passes the request on. Note that the media servers are controlled remotely from the main SIP which can cause additional complexity and call setup delay. Figure 13 displays the interaction between the different components and the exchanged messages.

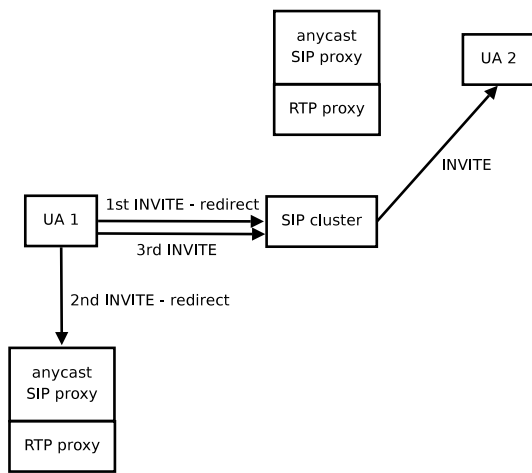


Fig. 13. Anycast “bootstrap” redirect service

1) *Call Flows:* In the following the call sequence when using an RTP proxy for NAT traversal is explained, see Figure 14

a) REGISTER:

1. UA1 sends a REGISTER request to the unicast IP address of the provider’s main SIP server where the UA1s Contact is saved in the location database. In case the UA is behind a NAT the SIP server also notes that the UA is behind a NAT.

b) INVITE:

2. UA1 sends an INVITE request to the main SIP server that checks if UA1 is behind a NAT. If so, then it checks if the combination Client-IP, RTPproxy IP is in the cache (location DB in memory). If the cache is empty then the SIP server redirects the request to a SIP server listening on an anycast IP address.
3. UA1 resends the INVITE request to the anycast SIP server which redirects the INVITE back to the main SIP server. The redirection URI in the

Contact header indicates a URI parameter with the unicast IP address of the co-located RTP proxy.

4. UA1 sends the INVITE to the main SIP server using as the Contact header the header received from the anycast SIP server which included the address of the main SIP server as well as a parameter indicating the location of the anycast SIP server which is also the address of the media server in the proximity of the user.
5. The main SIP server uses the information in the Contact header for selecting a media server that is subsequently used for relaying media. The message is record-routed to stay in the path for BYEs and forwards it to UA2.

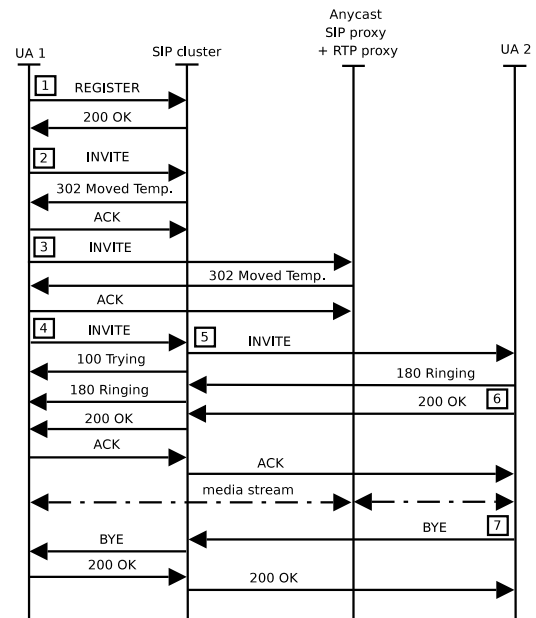


Fig. 14. Anycast “bootstrap” redirect service call flow

VI. EVALUATION OF METHODS

In this section we highlight the main advantages and disadvantages of each of the presented approaches.

A. Anycasting DNS Servers

- **Ease of deployment:** This method is simple to integrate at the system level and SIP messages are simply sent to a unicast IP address returned from DNS lookup.
- **Interoperability:** No extensions or additions are needed for SIP.
- **Proximity:** The proximity measurement may be impaired if a client uses a DNS resolver that is not located in its proximity as in that case using anycast

results in detecting a server that is in the proximity of the DNS server and not of the user.

- **Resilience:** This method is resilient against routing instabilities as the anycast traffic is limited to a short-lived UDP-based DNS transaction. However, DNS resolvers in SIP clients and DNS proxy servers are known to cache DNS information for quite long periods. If an anycast site fails and stops advertising its route, poor DNS clients will keep using an unavailable IP address.

B. Anycasting SIP Servers

- **Ease of deployment:** This method is simple to integrate at the system level and SIP messages are simply sent to the anycast IP address of the SIP servers.
- **Interoperability:** No extensions or additions are needed for SIP.
- **Proximity:** As the SIP requests are sent directly to the anycast address of the SIP servers, the users should end up using servers in their proximity
- **Resilience:** Most sensitive against routing instabilities making any usage scenarios requiring stateful operation difficult if not impossible to realize.

C. Anycast SIP Tunnels

- **Ease of deployment:** The provider's SIP servers will have to maintain IP tunnels to all used media servers this could become a configuration nightmare if there are a lot of media and SIP servers. Further, as the SIP servers have to remotely control the media servers, e.g., open a port or instruct it to handle incoming traffic in a certain manner the call set-up is delayed.
- **Interoperability:** No extensions or additions are needed for SIP.
- **Proximity:** As the SIP requests are sent directly to the anycast address of the SIP servers, the users should end up using servers in their proximity
- **Resilience:** As all related transport and SIP state information are saved at the SIP server routing instabilities do not cause any issue on the SIP level. Media servers are contacted over unicast addresses so they are not affected by routing instabilities.

D. Anycast "bootstrap" Redirect Server

- **Ease of deployment:** This method is simple to integrate at the system level and SIP messages are simply sent to anycast/unicast IP address of the SIP servers.

- **Interoperability:** The user agent will need to support redirection and not change the Contact header received from the anycast servers. However, it is often the case that SIP clients do not support redirection due to poor implementation or for policy reasons as redirection could be used for fraud when a user is lured to call a free number but then gets directed to a costly number for example. Also, call setup latency increases as the INVITE request is redirected from the main server to the media server and back to the main server.
- **Proximity:** As the SIP requests are sent directly to the anycast address of the SIP servers, the users should end up using servers in their proximity.
- **Resilience:** As all related transport and SIP state information are saved at the SIP server routing instabilities do not cause any issue on the SIP level. Media servers are contacted over unicast addresses so they are not affected by routing instabilities.

VII. SUMMARY AND FUTURE WORK

In this paper we investigated different approaches for using distributed media servers and enabling the users to choose media servers in their proximity. The different approaches have their pros and cons. In a second step we need to investigate the performance of the different approaches on a real live test bed and evaluate the different approaches with regard to their reliability and robustness.

IP anycast wastes the address space as the longest IP prefix is that can be announced using common routing protocols is /24 and it requires the VoIP providers to have the capability of announcing routing prefixes. This can make the deployment of IP anycast for small VoIP providers rather complex as it would require them to deploy routing protocols and apply for an autonomous system number. Therefore, using IP anycast as the basis for a distributed media server architecture is more appropriate for large providers or could be offered as an independent service in a similar manner to content distribution networks which offer the service of content distribution to content providers, e.g., news sites. As part of our future work we will consider the security aspects of separating the signaling and media processing capabilities into two independent services.

While this is still early work, we feel compelled to work in the future on advancing from proof-of-concept to a study based on a reasonable measurement data set. Such would allow us to provide stronger statements on efficiency of anycast to discover the closest relay, significance of routing instabilities, and routing convergence on network failures. Also we plan to continue studying the

architectural aspects, such as taking advantage of anycast for geographic dispersion, taking proximity of both call parties in account and studying security implications.

[17] H. Schulzrinne and B. Volz, "Dynamic Host Configuration Protocol (DHCPv6) Options for Session Initiation Protocol (SIP) Servers," RFC 3319 (Proposed Standard), Jul. 2003. [Online]. Available: <http://www.ietf.org/rfc/rfc3319.txt>

REFERENCES

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), Jun. 2002.
- [2] J. Rosenberg, R. Mahy, and P. Matthews, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)," draft-ietf-behave-turn-09, Internet Engineering Task Force, Internet Draft draft-ietf-behave-turn-09, Jul. 2008, work in progress.
- [3] K. Hedayat, N. Venna, P. Jones, A. Roychowdhury, C. SivaChelvan, and N. Stratton, "An Extension to the Session Description Protocol (SDP) for Media Loopback," draft-ietf-mmusic-media-loopback-10, Internet Engineering Task Force, Internet Draft draft-ietf-mmusic-media-loopback-10, Feb. 2009, work in progress.
- [4] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service," RFC 1546 (Informational), Nov. 1993. [Online]. Available: <http://www.ietf.org/rfc/rfc1546.txt>
- [5] H. Ballani and P. Francis, "Towards a global ip anycast service," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 301–312, 2005.
- [6] J. Abley and K. Lindqvist, "Operation of Anycast Services," RFC 4786 (Best Current Practice), Dec. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4786.txt>
- [7] J. Abley, "Hierarchical Anycast for Global Service Distribution," <http://www.isc.org/pubs/tn/isc-tn-2003-1.html>, 2003.
- [8] R. Engel, V. Peris, D. Saha, and E. Basturk, "Abstract using ip anycast for load distribution and server location," in *Global Internet*, 1998.
- [9] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271 (Draft Standard), Jan. 2006. [Online]. Available: <http://www.ietf.org/rfc/rfc4271.txt>
- [10] Z. M. Mao, R. Govindan, G. Varghese, and R. H. Katz, "Route flap damping exacerbates internet routing convergence," in *SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications*. New York, NY, USA: ACM, 2002, pp. 221–233.
- [11] H. Ballani, P. Francis, and S. Ratnasamy, "A Measurement-based Deployment Proposal for IP Anycast," in *Proc. of Internet Measurement Conference*, October 2006.
- [12] J. Postel, "Internet Control Message Protocol," RFC 792 (Standard), Sep. 1981, updated by RFCs 950, 4884. [Online]. Available: <http://www.ietf.org/rfc/rfc792.txt>
- [13] J. Pan, Y. T. Hou, and B. Li, "An overview of dns-based server selections in content distribution networks," *Comput. Netw.*, vol. 43, no. 6, pp. 695–711, 2003.
- [14] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131 (Draft Standard), Mar. 1997, updated by RFCs 3396, 4361. [Online]. Available: <http://www.ietf.org/rfc/rfc2131.txt>
- [15] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," RFC 3315 (Proposed Standard), Jul. 2003, updated by RFC 4361. [Online]. Available: <http://www.ietf.org/rfc/rfc3315.txt>
- [16] H. Schulzrinne, "Dynamic Host Configuration Protocol (DHCP-for-IPv4) Option for Session Initiation Protocol (SIP) Servers," RFC 3361 (Proposed Standard), Aug. 2002. [Online]. Available: <http://www.ietf.org/rfc/rfc3361.txt>