TRAM
Internet-Draft
Intended status: Standards Track
Expires: January 25, 2015
P. Patil
T. Reddy
D. Wing
Cisco
July 24, 2014

TURN Server Auto Discovery
draft-ietf-tram-turn-server-discovery-00

Abstract

Current Traversal Using Relays around NAT (TURN) server discovery mechanisms are relatively static and limited to explicit configuration. These are usually under the administrative control of the application or TURN service provider, and not the enterprise or the ISP, the network in which the client is located. Enterprises and ISPs wishing to provide their own TURN servers need auto discovery mechanisms that a TURN client could use with no or minimal configuration. This document describes two such mechanisms for TURN server discovery.

Table of Contents

1. Introduction

TURN [RFC5766] is a protocol that is often used to improve the connectivity of P2P applications. By providing a relay service, TURN ensures that a connection can be established even when one or both sides is incapable of a direct P2P connection. It is an important building block for interactive, real-time communication using audio, video, collaboration etc. While TURN services are extensively used today, the means to auto discover TURN servers do not exist. TURN clients are usually explicitly configured with a well known TURN server. To allow TURN applications operate seamlessly across different types of networks and encourage the use of TURN without the need for manual configuration, it is important that there exists an auto discovery mechanism for TURN services. WebRTC usages and

related extensions, which are mostly based on web applications, need this immediately.

This document describes two discovery mechanisms. The reason for providing two mechanisms is to maximize the opportunity for discovery, based on the network in the which the TURN client sees itself.

- o A resolution mechanism based on straightforward Naming Authority Pointer (S-NAPTR) resource records in the Domain Name System (DNS). [RFC5928] describes details on retrieving a list of server transport addresses from DNS that can be used to create a TURN allocation.

- o A mechanism based on anycast address for TURN.

In general, if a client wishes to communicate using one of its interfaces using a specific IP address family, it SHOULD query the TURN server(s) that has been discovered for that specific interface and address family. How to select an interface and IP address family, is out of the scope of this document.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Discovery Procedure

A TURN client that implements the auto discovery algorithm MUST proceed with discovery in the following order:

1. Local Configuration : Local or manual configuration should be tried first, as it may be an explicit preferred choice of a user. An implementation MAY give the user an opportunity (e.g., by means of configuration file options or menu items) to specify a TURN server for every address family.

2. Service Resolution : The TURN client attempts to perform TURN service resolution using the DNS domain name that the host belongs to OR the hosts' global IP address. The TURN client will attempt to do this for each combination of interface and address family. The retrieved DNS domain names OR IP addresses are then used for NAPTR lookups.

3. Anycast : Send TURN allocate request to the assigned TURN anycast request for each combination of interface and address family.

While it is expected that Step-3 be performed if Step-2 fails, an implementation may choose to perform steps 2 and 3 in parallel.

4. Discovery using Service Resolution

This mechanism is performed in two steps:

1. A DNS domain name is retrieved for each combination of interface and address family.

2. Retrieved DNS domain names are then used for S-NAPTR lookups as per [RFC5928]. Further DNS lookups may be necessary to determine TURN server IP address(es).

On hosts with more than one interface or address family (IPv4/v6), the TURN server discovery procedure has to be run for each combination of interface and address family.

4.1. Retrieving Domain Name

The domain, in which the client is located, can be determined using one of the techniques provided below. An implementation can choose to use any or all techniques.

Implementations may allow the user to specify a default name that is used if no specific name has been configured. Other means of retrieving domain names may be used, which are outside the scope of this document e.g. local configuration.

4.1.1. DHCP

DHCP can be used to determine the domain name related to an interface's point of network attachment. Network operators may provide the domain name to be used for service discovery within an access network using DHCP. [RFC5986] defines DHCP IPv4 and IPv6 access network domain name options to identify a domain name that is suitable for service discovery within the access network. [RFC2132] defines the DHCP IPv4 domain name option. While this option is less suitable, it still may be useful if the option defined in [RFC5986] is not available.

For IPv6, the TURN server discovery procedure MUST try to retrieve DHCP option 57 (OPTION_V6_ACCESS_DOMAIN). If no such option can be retrieved, the procedure fails for this interface. For IPv4, the TURN server discovery procedure MUST try to retrieve DHCP option 213 (OPTION_V4_ACCESS_DOMAIN). If no such option can be retrieved, the procedure SHOULD try to retrieve option 15 (Domain Name). If neither option can be retrieved the procedure fails for this interface. If a

result can be retrieved it will be used as an input for S-NAPTR resolution.

4.1.2. IP Address

Typically, but not necessarily, the DNS domain name is the domain name in which the client is located, i.e., a PTR lookup on the client's IP address (according to [RFC1035], Section 3.5 for IPv4 or [RFC3596], Section 2.5 for IPv6) would yield a similar name. However, due to the widespread use of Network Address Translation (NAT), the client MAY need to determine its public IP address using mechanisms described in [RFC7216].

4.1.3. From own Identity

A TURN client could also wish to extract the domain name from its own identity i.e canonical identifier used to reach the user.

Example

```
SIP   : 'sip:alice@example.com'
JID   : 'alice@example.com'
email : 'alice@example.com'
```

'example.com' is retrieved from the above examples.

The means to extract the domain name may be different based on the type of identifier and is outside the scope of this document.

4.2. Resolution

Once the TURN discovery procedure has retrieved domain names, the resolution mechanism described in [RFC5928] is followed. An S-NAPTR lookup with 'RELAY' application service and the desired protocol tag is made to obtain information necessary to connect to the authoritative TURN server within the given domain.

In the example below, for domain 'example.net', the resolution algorithm will result in IP address, port, and protocol tuples as follows:

```
example.net.
   IN NAPTR 100 10 "" RELAY:turn.udp "" example.net.

   example.net.
   IN NAPTR 100 10 S RELAY:turn.udp "" _turn._udp.example.net.

   _turn._udp.example.net.
   IN SRV   0 0 3478 a.example.net.

   a.example.net.
   IN A     192.0.2.1
```

Order	Protocol	IP address	Port
1	UDP	192.0.2.1	3478

If no TURN-specific S-NAPTR records can be retrieved, the discovery procedure fails for this domain name (and the corresponding interface and IP protocol version). If more domain names are known, the discovery procedure may perform the corresponding S-NAPTR lookups immediately. However, before retrying a lookup that has failed, a client MUST wait a time period that is appropriate for the encountered error (NXDOMAIN, timeout, etc.).

4.2.1. SOA

If no TURN-specific S-NAPTR records can be retrieved using the previous step, additional steps described in this section have to be followed. First, an SOA record for the "reverse zone" i.e., the zone in the in-addr.arpa. or ip6.arpa. domain that contains the IP address(s) in question, has to be retrieved. IP addresses can be determined, if not done already, as described in Section 4.1.2.

A sample SOA record could be:

```
100.51.198.in-addr.arpa
IN  SOA dns1.isp.example.net.   hostmaster.isp.example.net. (
                           1         ; Serial
                      604800         ; Refresh
                       86400         ; Retry
                     2419200         ; Expire
                      604800 )       ; Negative Cache TTL
```

If this lookup fails, the discovery procedure is aborted without a result.

Once the SOA record is available, the discovery procedure extracts the MNAME field, i.e., the responsible master name server from the SOA record. An example MNAME could be: dns1.isp.example.net. Then, an S-NAPTR lookup as specified in the previous step Section 4.2 is performed on this MNAME to discover the TURN service. If no TURN-specific S-NAPTR records can be retrieved, the discovery procedure fails for this domain name (and the corresponding interface and IP protocol version).

5. Discovery using Anycast

IP anycast is an elegant solution for TURN service discovery. A packet sent to an anycast address is delivered to the "topologically nearest" network interface with the anycast address. Using the TURN anycast address, the only two things that need to be deployed in the network are the two things that actually use TURN.

When a client requires TURN services, it sends a TURN allocate request to the assigned anycast address. The TURN anycast server responds with a 300 (Try Alternate) error as described in [RFC5766]; The response contains the TURN unicast address in the ALTERNATE-SERVER attribute. For subsequent communication with the TURN server, the client uses the responding server's unicast address. This has to be done because two packets addressed to an anycast address may reach two different anycast servers. The client, thus, also needs to ensure that the initial request fits in a single packet. An implementation may choose to send out every new request to the anycast address to learn the closest TURN server each time.

6. Deployment Considerations

6.1. Mobility and Changing IP addresses

A change of IP address on an interface may invalidate the result of the TURN server discovery procedure. For instance, if the IP address assigned to a mobile host changes due to host mobility, it may be required to re-run the TURN server discovery procedure without relying on earlier gained information. New requests should be made to the newly learned TURN servers learned after TURN discovery re-run. However, if an earlier learned TURN server is still accessible using the new IP address, procedures described for mobility using TURN defined in [I-D.wing-mmusic-ice-mobility] can be used for ongoing streams.

7. IANA Considerations

7.1. Anycast

IANA should allocate an IPv4 and an IPv6 well-known TURN anycast address. 192.0.0.0/24 and 2001:0000::/48 are reserved for IETF Protocol Assignments, as listed at

<http://www.iana.org/assignments/iana-ipv4-special-registry/> and

<http://www.iana.org/assignments/iana-ipv6-special-registry/>

8. Security Considerations

In general, it is recommended that a TURN client authenticate with the TURN server to identify a rouge server. [I-D.petithuguenin-tram-turn-dtls] can be potentially used by a client to validate a previously unknown server.

8.1. Service Resolution

The primary attack against the methods described in this document is one that would lead to impersonation of a TURN server. An attacker could attempt to compromise the S-NAPTR resolution. Security considerations described in [RFC5928] are applicable here as well.

In addition to considerations related to S-NAPTR, it is important to recognize that the output of this is entirely dependent on its input. An attacker who can control the domain name can also control the final result. Because more than one method can be used to determine the domain name, a host implementation needs to consider attacks against each of the methods that are used.

If DHCP is used, the integrity of DHCP options is limited by the security of the channel over which they are provided. Physical security and separation of DHCP messages from other packets are commonplace methods that can reduce the possibility of attack within an access network; alternatively, DHCP authentication [RFC3188] can provide a degree of protection against modification. When using DHCP discovery, clients are encouraged to use unicast DHCP INFORM queries instead of broadcast queries which are more easily spoofed in insecure networks.

8.2. Anycast

In a network without any TURN server that is aware of the TURN anycast address, outgoing TURN requests could leak out onto the external Internet, possibly revealing information.

Using an IANA-assigned well-known TURN anycast address enables border gateways to block such outgoing packets. In the default-free zone, routers should be configured to drop such packets. Such configuration can occur naturally via BGP messages advertising that no route exists to said address.

Sensitive clients that do not wish to leak information about their presence can set an IP TTL on their TURN requests that limits how far they can travel into the public Internet.

9. Acknowledgements

Discovery using Service Resolution described in Section 4 of this document was derived from similar techniques described in ALTO Server Discovery [I-D.ietf-alto-server-discovery] and [I-D.kist-alto-3pdisc].

10. References

10.1. Normative References

[I-D.petithuguenin-tram-turn-dtls]
Petit-Huguenin, M. and G. Salgueiro, "Datagram Transport Layer Security (DTLS) as Transport for Traversal Using Relays around NAT (TURN)", draft-petithuguenin-tram-turn-dtls-00 (work in progress), January 2014.

[I-D.wing-mmusic-ice-mobility]
Wing, D., Reddy, T., Patil, P., and P. Martinsen, "Mobility with ICE (MICE)", draft-wing-mmusic-ice-mobility-07 (work in progress), June 2014.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[RFC5766] Mahy, R., Matthews, P., and J. Rosenberg, "Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN)", RFC 5766, April 2010.

[RFC5928] Petit-Huguenin, M., "Traversal Using Relays around NAT (TURN) Resolution Mechanism", RFC 5928, August 2010.

[RFC5986] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", RFC 5986, September 2010.

[RFC7216] Thomson, M. and R. Bellis, "Location Information Server (LIS) Discovery Using IP Addresses and Reverse DNS", RFC 7216, April 2014.

10.2. Informative References

[I-D.ietf-alto-server-discovery]
Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., and S. Yongchao, "ALTO Server Discovery", draft-ietf-alto-server-discovery-10 (work in progress), September 2013.

[I-D.kist-alto-3pdisc]
Kiesel, S., Krause, K., and M. Stiemerling, "Third-Party ALTO Server Discovery (3pdisc)", draft-kist-alto-3pdisc-05 (work in progress), January 2014.

[RFC3188] Hakala, J., "Using National Bibliography Numbers as Uniform Resource Names", RFC 3188, October 2001.

Appendix A. Change History

[Note to RFC Editor: Please remove this section prior to publication.]

A.1. Change from draft-patil-tram-serv-disc-00 to -01

- Added IP address (Section 4.1.2) and Own identity (4.1.3) as new means to obtain domain names
- New Section 4.2.1 SOA (inspired by draft-kist-alto-3pdisc)
- 300 (Try Alternate) response for Anycast

Authors' Addresses

Prashanth Patil
Cisco Systems, Inc.
Bangalore
India

Email: praspati@cisco.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: tireddy@cisco.com

Dan Wing
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, California 95134
USA

Email: dwing@cisco.com