TURN - Traversal Using Relays around NAT

Traversal Using Relays around NAT (TURN) is a protocol that assists in traversal of network address translators (NAT) or firewalls for multimedia applications. It may be used with the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). It is most useful for clients on networks masqueraded by symmetric NAT devices. TURN does not aid in running servers on well known ports in the private network through a NAT; it supports the connection of a user behind a NAT to only a single peer, as in telephony, for example.

TURN is specified by RFC 5766. An update to TURN for IPv6 is specified in RFC 6156. The TURN URI scheme is documented in RFC 7065.

The process begins when a client computer wants to contact a peer computer for a data transaction, but cannot do so due to both, client and peer, being behind respective NATs. If STUN is not an option because one of the NATs is a symmetric NAT (a type of NAT known to be non-STUN compatible), TURN must be used.

First, the client contacts a TURN server with an "Allocate" request. The Allocate request asks the TURN server to allocate some of its resources for the client so that it may contact a peer. If allocation is possible, the server allocates an address for the client to use as a relay, and sends the client an "Allocation Successful" response, which contains an "allocated relayed transport address" located at the TURN server.

Second, the client sends in a CreatePermissions request to the TURN server to create a permissions check system for peer-server communications. In other words, when a peer is finally contacted and sends information back to the TURN server to be relayed to client, the TURN server uses the permissions to verify that the peer-to-TURN server communication is valid.

After permissions have been created, the client has two choices for sending the actual data, (1) it can use the Send mechanism, or (2) it can reserve a channel using the ChannelBind request. The Send mechanism is more straightforward, but contains a larger header, 36 bytes, that can substantially increase the bandwidth in a TURN relayed conversation. By contrast, the ChannelBind method is lighter: the header is only 4 bytes, but it requires a channel to be reserved which needs to be periodically refreshed, among other considerations.

Using either method, Send or channel binding, the TURN server receives the data from the client and relays it to the peer using UDP datagrams, which contain as their Source Address the "Allocated Relayed Transport Address". The peer receives the data and responds, again using a UDP datagram as the transport protocol, sending the UDP datagram to the relay address at the TURN server.

The TURN server receives the peer UDP datagram, checks the permissions and if they are valid, forwards it to the client.

This process gets around even symmetric NATs because both the client and peer can at least talk to the TURN server, which has allocated a relay IP address for communication.

While TURN is more robust than STUN in that it assist in traversal of more types of NATs, a TURN communication relays the entire communication through the server requiring far more server bandwidth than the STUN protocol, which typically only resolves the public facing IP address and relays the information to client and peer for them to use in direct communication. For this reason, the ICE protocol mandates STUN usage as a first resort, and TURN usage only when dealing with symmetric NATs or other situations where STUN cannot be used.

Source: https://en.wikipedia.org/wiki/Traversal_Using_Relays_around_NAT

Related Links:

- CSTA - Computer-supported telecommunications applications
- LLDP - Link Layer Discovery Protocol
- PoP - Point of Presence
- RTP - Real-Time Transport Protocol
- SBC - Session Border Controller
- SRTP - Secure Real-Time Transport Protocol
- STUN - Simple traversal of UDP over NAT