

TURN URI and Resolution Mechanism

Marc Petit-Huguenin

draft-petithuguenin-behave-turn-uri

TURN URI

- Syntax inspired by DNS URI:
 - `turn://example.org`
 - `turns://example.org`
 - `turn://example.org:3178`
 - `turn://192.0.2.1:3178`
 - `turn://example.org?transport=UDP`

Resolution Mechanism

- Inspired by RFC 3263 but adapted to use RFC 3958 (S-NAPTR)
- Domain administrator can order the TURN protocols by preference.
- Can be used for remote hosting.

Example

- example.com
IN NAPTR 100 10 "" "RELAY:turn.udp" "" datagram.example.com.
IN NAPTR 200 10 "" "RELAY:turn.tcp:turn.tls" "" stream.example.com.
- datagram.example.com.
IN NAPTR 100 10 "S" "RELAY:turn.udp" "" _udp._turn.example.com.
- stream.example.com.
IN NAPTR 100 10 "A" "RELAY:turn.tls" "" a.example.com.
IN NAPTR 200 10 "S" "RELAY:turn.tcp" "" _tcp._turn.example.com.
- _udp._turn.example.com.
IN SRV 0 0 5000 a.example.com.
- _tcp._turn.example.com.
IN SRV 0 0 5000 a.example.com.
- a.example.com.
IN A 192.0.2.1

URI extensibility?

- The only component that is currently extensible in a TURN URI is the transport parameter. No other parameter can be added.
- The problem of permitting additional parameters is the behavior for resolvers that does not understand them.

Additional ways to select the servers?

- Optional capabilities for TURN servers:
 - IPv6 allocation
 - Preserve bit
 - Allocate even port bit
 - Allocate next port bit
- Instead of trying all servers returned by SRV RR until one supports the capabilities requested, the server capabilities can be provisioned in the NAPTR RR.

Next steps

- WG item?