Network Working Group
Internet-Draft
Expires: December 23, 2002

F. Parent
M. Blanchet
Viagenie inc.
June 24, 2002

TSP-TEREDO: Stateful IPv6 over IPv4 Tunnels with NAT using TSP and TEREDO
draft-parent-blanchet-ngtrans-tsp-teredo-00.txt

Status of this Memo

Copyright Notice

Abstract

Teredo [3] is a stateless mechanism to encapsulate IPv6 traffic in IPv4 when there is an IPv4 NAT between the tunnel endpoints. This document enhances Teredo by providing a stateful IPv6 in IPv4 connexion which enables additional features to be used, like permanent IPv6 address or prefixes. It uses the Tunnel Setup Protocol (TSP) [2] to negociate the parameters of the tunnel and identify the NAT translation map. TSP also enables negociation and establishment of prefixes, routing, DNS delegation and other parameters related to the tunnel.

Table of Contents

1. Introduction

Teredo [3] describes a service that enables nodes located behind one or several IPv4 NATs to obtain IPv6 connectivity by tunneling packets over UDP.

This document describes how Teredo and TSP can be used together to provide new services to nodes behind an IPv4 NAT such as permanent IPv6 address or prefixes allocation, routing services, DNS delegation and other parameters related to the tunnel.

2. TSP-Teredo Operation

In order to offer services such as permanent IPv6 address or prefix to clients, the Teredo server must work in stateful mode, where authentication phase is required prior to allocating a prefix to the client. This allows to tie an IPv6 prefix to an identity chosen by the client. This is the scheme proposed in TSP [2] and this document describes how TSP can be used in the Teredo context.

In stateless Teredo [3], the initial traffic sent by the client is an IPv6 router solicitation to the Teredo server. By its stateless nature, the Teredo server offers IPv6 connectivity to any client.

In the first phase of Teredo operation, the server obtains the IPv4 mapped (by NAT) address and UDP port of the client from the IPv6 router solicitation received from that client. Using this information, the client prefix is created from the Teredo global prefix space.

This document proposes using TSP to allocate the IPv6 address to the client. The address allocated is taken from the global IPv6 address space available to the TSP-Teredo server.

The syntax of the protocol is given in Section 3. Tunnel Setup Protocol [2] proposes a two phase process: In the first phase, the client authenticates to the tunnel server. If the authentication is successful, the client can request a tunnel establishement with a permanent (or temporary) prefix. The client prefix assignment is done during the TSP session.

In TSP-Teredo mode, the TSP protocol is transported using UDP over IPv4, and the UDP port is the Teredo/TSP server port (to be defined). Upon successful authentication and address allocation, the IPv4 mapped (by NAT) address and UDP port of the client from the TSP packet are used to establish an IPv6 over UDP over IPv4 tunnel between the server and client.

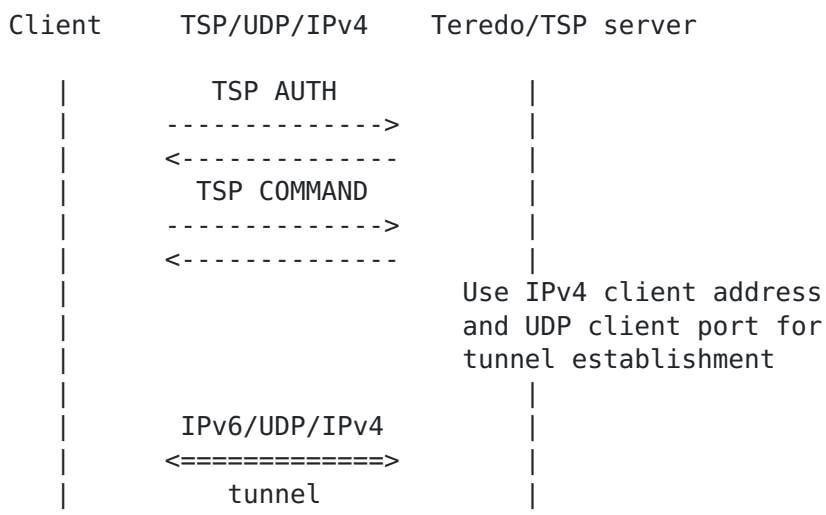

Figure 1: Client to server initial negociation

3. TSP Profile for Teredo

The TSP profile uses a defined DTD (Appendix A) for the XML format of the message. The DTD contains the description of the tunnel XML message. This message is used by the TSP Teredo compliant host and server to provide the necessary information to establish an IPv6 in UDPv4 tunnel.

A complete description of the protocol syntax can be found in TSP [2]. Examples of the client/server exchange are given in Section 5.

3.1 Element 'tunnel'

The TSP message is composed of a 'tunnel' element that contains 0 or one server, client or broker elements. The 'tunnel' element is defined with 3 attributes which describes the 'action' requested in the message, the 'type' of tunnel requested, and the 'lifetime' of that tunnel.

The following sections define the different 'tunnel' element attributes

3.1.1 Attribute 'action'

Three values are possible for this attribute: create, delete and info.

create: Sent by the client to request a new tunnel or update an existing tunnel.

info: Sent by the client to request current properties of an existing tunnel. Also contains tunnel information sent by server to client

delete: Sent by client to remove an existing tunnel on the server.

3.1.2 Attribute 'type'

Identifies the type of tunnel requested.

v6v4: request/allocate an IPv6 in IPv4 [5] tunnel

v6udpv4: request/allocate an IPv6 in UDPv4 tunnel

v6any: request a tunnel of any type supported. Can only be used when requesting a tunnel

3.1.3 Attribute 'lifetime'

Length of time (minutes) when the tunnel is valid. This is not the prefix valid or preferred lifetime. Default is 1440 minutes, or 24 hours.

3.2 Element 'server'

This element is used to describe the server tunnel endpoint. The 'server' element contains 2 elements: 'address' and 'router'. The 'address' element is used to send both IPv4 and IPv6 addresses of the server's tunnel endpoint. The 'router' element may be present to provide information for the routing method choosen by the client.

3.3 Element 'client'

This element is used to describe the client parameters and will be used by the server to create the appropriate tunnel. This is the only element sent by a client to the server.

The 'client' element MUST contain one or more 'address' elements. The server will then return the IPv6 address endpoint and domain name

inside the 'client' element when the tunnel is created or updated.

The 'client' element MAY contain one 'router' element to ask for a prefix delegation. The 'router' element contains the 'protocol' attribute which specify the routing method to be use between the server and the client. If no attribute is specified the the routing will use static routes.

3.4 Element 'router'

This element is used by the client to request a prefix delegation. The 'router' element can be specified with an attribute to specify the routing protocol to be used between the client and server.

The 'router' element may contain the following elements:

prefix: Used by the client to request a prefix length. Used by the server to specify the prefix allocated.

dns_server: Used by the client to request DNS delegation for the requestd prefix.

as: Used when BGP routing is negociated.

3.5 Element 'dns_server'

This element is used for DNS delegation of the allocated prefix. The 'dns_server' is composed of one or more 'address' elements that specify the name servers used for the delegation.

3.6 Element 'prefix'

The 'prefix' element has a 'length' attribute that indicates the prefix length. When sent by the client, this element is used to specify the desired prefix length to the server. When sent by the server, both the prefix and prefix length are specified for the client configuration.

3.7 Element 'address'

In this element, the attribute 'type' is used to specify whether this element represents an IPv4 address, an IPv6 address or a domain name. The attribute 'length' represent the prefix length or netmask of the address, if applicable.

4. Tunnel encapsulation negociation

Assuming that the Teredo/TSP server supports IPv6 in IPv4 tunnels [5] in addition to IPv6 in UDPv4 , the TSP-Teredo server can negociate with the client on the tunnel that will be established.

During the TSP command phase, the Teredo/TSP server is able to detect if the client is behind a NAT by comparing the IPv4 address of the client inside the TSP protocol and the source address of the IPv4 header.

As shown in Figure 2, the client sends its IPv4 address and the tunnel type requested to the server when requesting a tunnel (command phase). In the example here, the client requested a tunnel type of 'v6any' (did not specify the encapsulation type).

The server is then able to compare the client IPv4 address from the received packet to the client IPv4 address in the TSP protocol. The server will offer a IPv6 in UDPv4

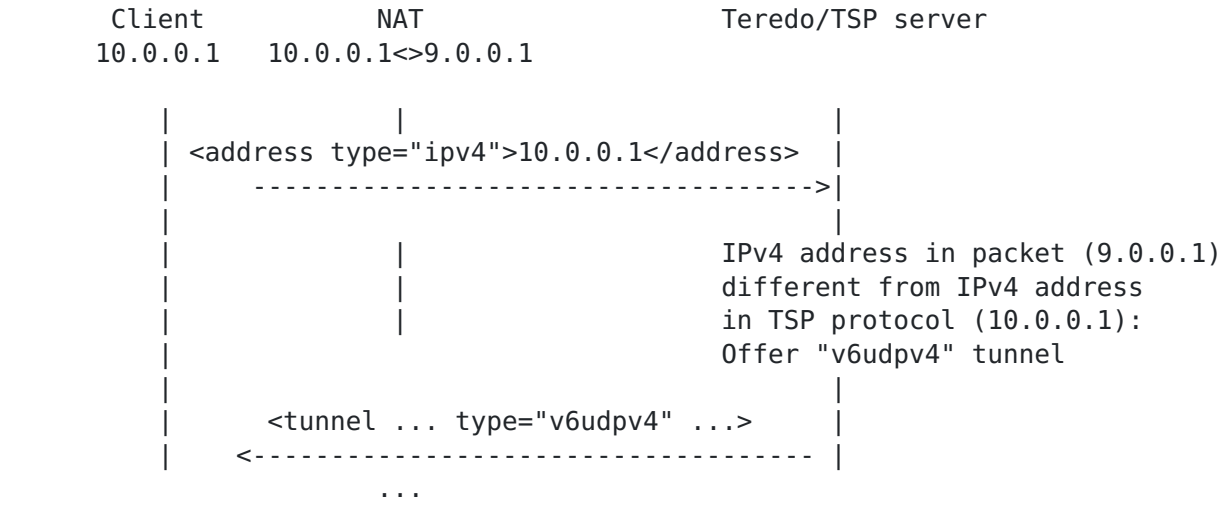

Figure 2: Server detecting if client behind NAT

5. Teredo/TSP client and server negociation examples

The following example demonstrates a client connecting to a Teredo/ TSP server to request a tunnel creation.

In the first phase, the server announces its capabilities (can provide both IPv6 in IPv4 and IPv6 in UDP in IPv4 tunnels) and the client authenticates.

In the second phase, the client requests a tunnel creation to transport IPv6 inside IPv4 without specifing the exact encapsulation (v6any) (the client may not know if it is behind a NAT or not). The client also sends its IPv4 address to the server. At this stage, the server is able to determine whether the client is behind a NAT by comparing the IPv4 address of the client inside the TSP protocol and the source address of the IPv4 header.

In the example below, the server detected that the client is behind a NAT so a v6 over UDPv4 tunnel is negociated to the client (offered tunnel type is v6udpv4).

```
     -- Successful TCP Connection --
     C:VERSION=1.1 CR LF

   remove attribute. add tunnel=v6udpv4
     S:CAPABILITY TUNNEL=V6V4 TUNNEL=V6UDPV4 AUTH=DIGEST-MD5 CR LF
     C:AUTHENTICATE ... CR LF
     S:200 Authentication successful CR LF

     C:Content-length: ... CR LF
       <tunnel action="create" type="v6any">
          <client>
              <address type="ipv4">10.1.1.1</address>
          </client>
       </tunnel> CR LF
     S: Content-length: ... CR LF
        200 OK CR LF
        <tunnel action="info" type="v6udpv4" lifetime="1440">
          <server>
             <address type="ipv4">206.123.31.114</address>
             <address type="ipv6">3ffe:b00:c18:ffff:0000:0000:0000:0000</
address>
          </server>
          <client>
             <address type="ipv4">10.1.1.1</address>
             <address type="ipv6">3ffe:b00:c18:ffff::0000:0000:0000:0001</
address>
          </client>
        </tunnel> CR LF
```

In the next example, the server determines that the client is not behind a NAT so the client is offered a standard IPv6 in IPv4 tunnel [ref].

```
     -- Successful TCP Connection --
     C:VERSION=1.1 CR LF
     S:CAPABILITY TUNNEL=V6V4 TUNNEL=V6UDPV4 AUTH=DIGEST-MD5 CR LF
     C:AUTHENTICATE ... CR LF
     S:200 Authentication successful CR LF

     C:Content-length: ... CR LF
       <tunnel action="create" type="v6any">
          <client>
              <address type="ipv4">9.1.1.1</address>
          </client>
       </tunnel> CR LF
     S: Content-length: ... CR LF
        200 OK CR LF
        <tunnel action="info" type="v6v4" lifetime="1440">
          <server>
             <address type="ipv4">206.123.31.114</address>
             <address type="ipv6">3ffe:b00:c18:ffff:0000:0000:0000:0000</
address>
          </server>
          <client>
             <address type="ipv4">9.1.1.1</address>
             <address type="ipv6">3ffe:b00:c18:ffff::0000:0000:0000:0001</
address>
          </client>
        </tunnel> CR LF
```

This example shows a client requesting a tunnel and an 48-bit IPv6 prefix. The server detected that the client is behind a NAT so a Teredo tunnel is negociated to the client (offered tunnel type is v6udpv4).

```
     -- Successful TCP Connection --
     C:VERSION=1.1 CR LF
     S:CAPABILITY TUNNEL=V6V4 TUNNEL=V6UDPV4 AUTH=DIGEST-MD5 CR LF
     C:AUTHENTICATE ... CR LF
     S:200 Authentication successful CR LF

     C:Content-length: ... CR LF
       <tunnel action="create" type="v6v4" type="v6udpv4">
          <client>
              <address type="ipv4">10.1.1.1</address>
              <router>
               <prefix length="48"/>
              </router>
          </client>
       </tunnel> CR LF
     S: Content-length: ... CR LF
        200 OK CR LF
        <tunnel action="info" type="v6udpv4" lifetime="1440">
          <server>
             <address type="ipv4">206.123.31.114</address>
             <address type="ipv6">3ffe:b00:c18:ffff:0000:0000:0000:0000</
address>
          </server>
          <client>
             <address type="ipv4">10.1.1.1</address>
             <address type="ipv6">3ffe:b00:c18:ffff::0000:0000:0000:0001</
address>
             <router>
              <prefix length="48">3ffe:0b00:c18:1234::</prefix>
             </router>
          </client>
        </tunnel> CR LF
```

6. Differences between TSP-Teredo and Teredo

This proposal describes how Teredo can use TSP to provide new services to clients. This section describes some of the new features and differences in Teredo with TSP:

- Client can be a host or a router. As a router, the client can receive an IPv6 prefix from the server.

- The IPv6 address (and prefix) assigned to the client is taken from the global address space available to the service provider. There is no need to embed the IPv4 and the UDP port number inside the IPv6 address assigned to the client.

- The client IPv6 address and prefix are fixed. This has many

benefits such as permitting the client to deploy services that require stable IPv6 addresses.

- If the NAT mapping for the client change, the tunnel needs to be re-established through TSP. But the client IPv6 address does not change.

- The client IPv6 address doesn't contain information that can be considered sensitive (NAT mapping of the client, firewall state).

- Client IPv6 traffic is always routed to its associated TSP/Teredo server. This is similar to an environment where clients dialup to the enterprise NAS. This may lead to sub-optimal IPv6 routing compared with stateless Teredo where the IPv6 traffic is routed to the closest Teredo relay.

- Client is authenticated.

7. Security considerations

TBD

References

[1] Blanchet, M., "IPv6 over IPv4 profile for Tunnel Setup Protocol (TSP)", draft-vg-ngtrans-tsp-v6v4profile-00 (work in progress), July 2001.

[2] Blanchet, M., "Tunnel Setup Protocol (TSP)", draft-vg-ngtrans-tsp-00 (work in progress), July 2001.

[3] Huitema, C., "Teredo: Tunneling IPv6 over UDP through NATs", draft-ietf-ngtrans-shipworm-05 (work in progress), February 2002.

[4] Durand, A., Fasano, P., Guardini, I. and D. Lento, "IPv6 Tunnel Broker", RFC 3053, January 2001.

[5] Gilligan, R. and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 2893, August 2000.

Authors' Addresses

Florent Parent
Viagenie inc.
2875 boul. Laurier, bureau 300
Sainte-Foy, QC G1V 2M2
Canada

Phone: +1 418 656 9254
EMail: Florent.Parent@viagenie.qc.ca
URI: http://www.viagenie.qc.ca/

Marc Blanchet
Viagenie inc.
2875 boul. Laurier, bureau 300
Sainte-Foy, QC G1V 2M2
Canada

Phone: +1 418 656 9254
EMail: Marc.Blanchet@viagenie.qc.ca
URI: http://www.viagenie.qc.ca/

Appendix A. IPv6 over UDPv4 tunnel DTD

```
<?xml version="1.0"?>

<!DOCTYPE tunnel  [

<!ELEMENT tunnel        (server?,client?,broker?)>

  <!ATTLIST tunnel action   (create|info|list)   #REQUIRED >
  <!ATTLIST tunnel type     (v6v4|v6udpv4|v6any) #REQUIRED >
  <!ATTLIST tunnel lifetime CDATA              "1440"    >

<!ELEMENT server        (address+,router?)>

<!ELEMENT client        (address+,router?)>

<!ELEMENT broker        (adress+)>

<!ELEMENT router        (prefix?,dns_server?,as?)>
  <!ATTLIST router protocol (rip|bgp) "">

<!ELEMENT dns_server    (address+)>

<!ELEMENT as EMPTY>
  <!ATTLIST as number CDATA #REQUIRED>

<!ELEMENT prefix        (#PCDATA)>
  <!ATTLIST prefix length CDATA #REQUIRED>

<!ELEMENT address       (#PCDATA)>
  <!ATTLIST address type (ipv4|ipv6|dn) #REQUIRED>
  <!ATTLIST address length CDATA "">

]>
```

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.