Network Working Group
Internet-Draft
Intended status: Experimental
Expires: August 14, 2018

M. Rosenau
February 10, 2018

TCP ~~option~~ Option "request IPv6 connection" draft-rosenau-request-v6-option-00

Abstract

This document describes an idea for a header "option" for the Transmission Control Protocol (TCP).

The extension is used by dual-stack nodes to force the use of the IPv6 protocol instead of the IPv4 protocol when exchanging data over TCP.

Especially if one of the two nodes is a NAT64 or NAT46 router it makes sense not to use IPv4 but IPv6 if both nodes are dual-stack nodes.

This document also suggests to declare the usage of such a method mandatory for all nodes that use IPv4 addresses that will be assigned by RIRs in the future.

Doing so will make it impossible to (mis-) use IPv4 addresses assigned by RIRs to ISPs for IPv4-only nodes that do not support IPv6.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 14, 2018.

Commentaire [Med1]: The document does not include any security discussion.

Commentaire [Med2]: Not sure to understand this case.

Commentaire [Med3]: I don't see how this can be mandated.

1. Introduction

Because of the IPv4 address shortage the IPv6 protocol has been developed. Unfortunately many nodes in the internet are still IPv4-only and many ISPs use NATs to ~~establish connections to servers~~share addresses among subscribers.

Some ISPs use NAT46 to allow IPv4-only clients to connect to an IPv6-only server. In many cases both nodes would be able to establish an IPv6 connection instead of an IPv4 connection because the use of the NAT can be avoided in this case.

This document describes a TCP [RFC0793] header "option" which allows both nodes to negotiate the use of the IPv6 protocol instead of the IPv4 protocol.

2. Terminology

2.1. Keywords in capital letters

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119] and indicate requirement levels for compliant implementations.

2.2. client

In the context of this document a "client" is a node which is establishing a TCP connection to another node by sending the first TCP packet.

Commentaire [Med4]: Do you have some references?

Actually, what is deployed is to handle the other way around: IPv6-only clients talking to IPV4-only servers. You may refer to BEHAVE RFCs.

Commentaire [Med5]: This assumes that both peers are IPv6-capable, i.e.:
-DS-to-DS
-v6-to-DS or DS-to-v6
-v6-to-v6

I guess you are targeting the first case, because for the other cases, IPv6 will be used.

2.3. server

In the context of this document a "server" is a node which is waiting for a client to establish a TCP connection with that node.

2.4. NAT64, NAT46

A NAT64 is a node which allows an IPv6-only client to connect to an IPv4-only server by translating IPv4 to IPv6 packets and vice versa or by acting as IPv6 server and IPv4 client the same time forwarding the data from one TCP connection to the other one.

A NAT46 is a node allowing an IPv4-only client to connect to an IPv6-only server.

Note that according to the definitions above NAT64 and NAT46 nodes are both "clients" and "servers" the same time.

Commentaire [Med6]: Please refer to RFC6146.

The last part of the definition is not aligned with 6146.

Commentaire [Med7]: I would delete this text.

3. Theory of operation

3.1. Connection to an IPv4-only server

A connection to a server not supporting this extension looks the following way:

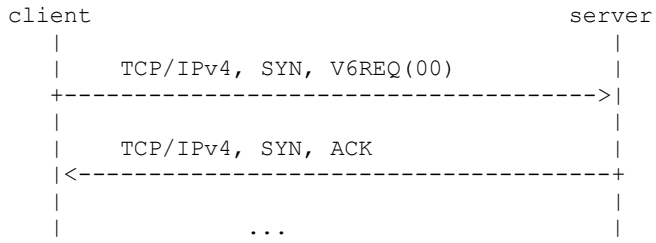

Figure 1: Request not understood by server

The client adds ~~an~~ the option described in this document to the "options"
field of the TCP header.

The server does not understand the option described in this document and it will ~~probably~~ ignore it. The TCP connection will be established "normally"; the option described in this document is not used.

3.2. IPv4 negotiated

If both the client and the server support this extension but both nodes decide to use IPv4 for the connection the connection looks the following way:

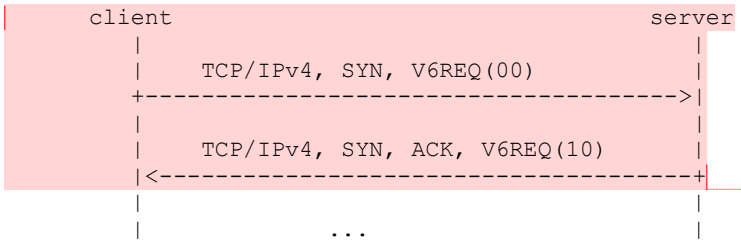

Figure 2: IPv4 negotiated

The server will send the "normal" (SYN/ACK) packet back to the client - as it is done for a normal TCP connection. It will add a certain option to the "options" field of the TCP header indicating that the option in the first packet is understood and supported by the server but it is not used for the current connection.

3.3. IPv6 negotiated

If the client and the server decide to use IPv6 for the connection the connection looks like this:

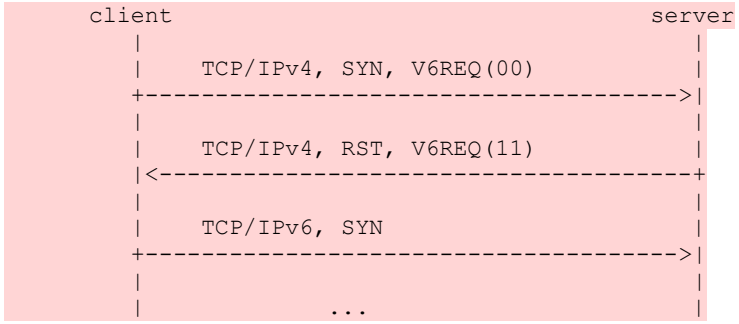

Figure 3: IPv6 negotiated

The server sends back a TCP packet containing the IPv6 address of the server and indicating that the IPv4 TCP connection is rejected.

Commentaire [Med8]: This assumes that the option is not striped by a middlebox and not manipulated by a misbehaving node.

Commentaire [Med9]: Why not using MPTCP to signal this?

You can refer to that spec for more details why advertising an address may be a complex task than expected.

The client will re-send an initial TCP packet (SYN) to the server using IPv6.

Commentaire [Med10]: This will lead to an extra delay to establish the session.

3.4. NAT scenario

There is even a situation where using this option in an IPv6 packet makes sense: When using with NAT64. Such a connection looks like this:

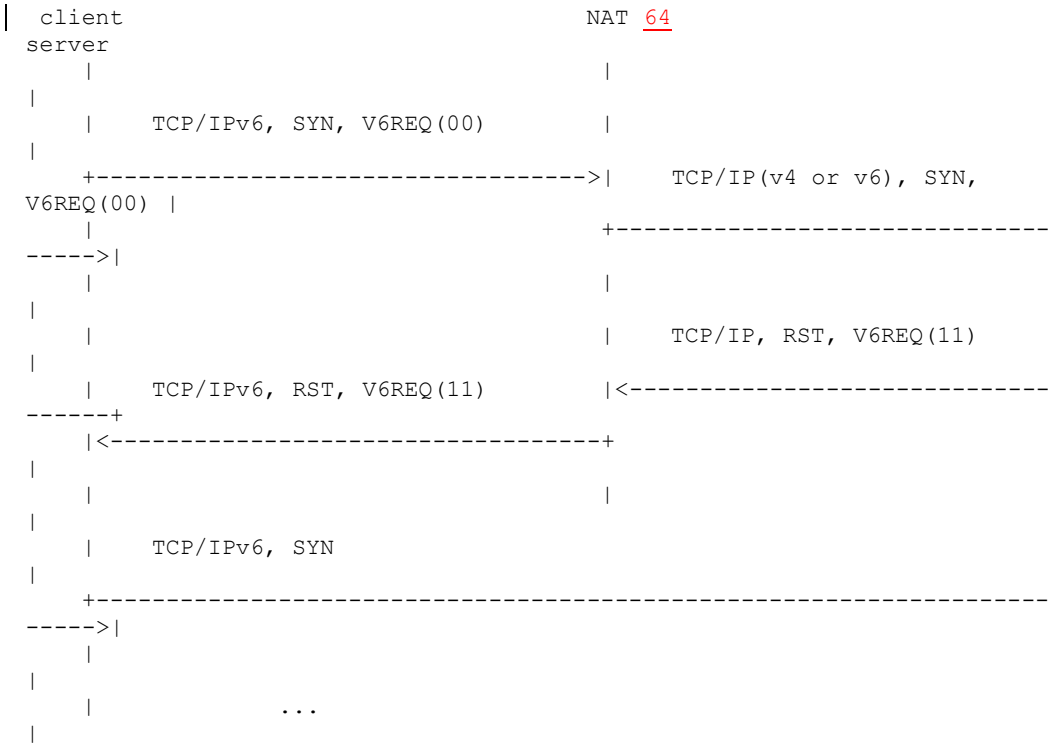

Figure 4: Using with NATs

The NAT64 sends back a TCP packet containing the IPv6 address of the server and indicating that the TCP connection to the NAT's address is rejected.

Commentaire [Med11]: If the server supports IPv6, why the client didn't use that address at the first place?

The client will re-send an initial TCP packet (SYN) directly to the server using IPv6 not using the NAT64.

4. Option formats

4.1. Request packet

In the initial TCP packet (the packet sent by the client to the server that has the SYN field set and the ACK field clear) the following TCP header option will be added by the client:

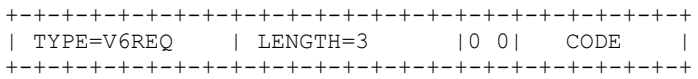

Figure 5: Request option format

The first octet of the option is the code "V6REQ" to be assigned by the IANA

The second octet of the option is the length of the option (as described in RFC 793 [RFC0793]). All codes described in this document use a length of 3 octets.

The high two bits of the third octet are zero and the low six bits are the code described below.

A TCP implementation MUST ignore this option (with the two upper bits in the third octet being zero) when it is found in any other packet but the initial TCP packet (SYN) sent from the client to the server.

If a TCP implementation does not understand the "CODE" (or the "LENGTH" field does not match the "CODE" field) it MUST ignore the option but it SHOULD add a "request not understood" option to the packet sent back to the client.

4.2. Informational response

If client and server decide to use IPv4 for the TCP connection the server answers as if the option was not present (SYN, ACK) but it adds the following option to the "options" field of the TCP header:

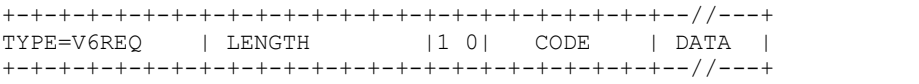

Figure 6: Informational response option format

The highest bit of the third octet is one and the bit below is set to zero. The low six bits are the code described below.

A TCP implementation MUST ignore this option (with the two upper bits in the third octet being one and zero) when it is found in any other packet but the TCP packet (SYN, ACK) sent as response to the first packet.

DATA is additional data depending on the "CODE" field.

If a TCP implementation does not understand the "CODE" (or the "LENGTH" field does not match the "CODE" field) it MUST ignore the option.

4.3. Force restart response

If client and server decide to use IPv6 for the TCP connection the server denies the IPv4-based connection and the client shall re-establish a connection via IPv6.

The server sends a packet with a special form that any TCP implementation that supports this option MUST understand:

The "source port" field contains the "destination port" of the initial packet and vice versa (just like in the case of a "normal" TCP response).

The "acknowledgement number" field contains the value of the "sequence number" field in the initial packet and vice versa. (Note that none of the two numbers is incremented.)

The "RST" bit MUST be set and "ACK" and "SYN" MUST be zero. The other control bits SHALL be zero.

The "options" field in the TCP header contains the following option:

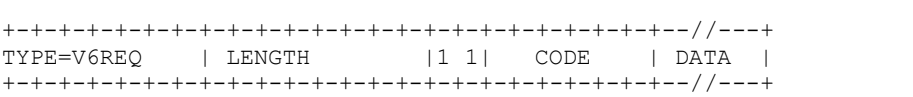

Figure 7: Force restart response option format

The highest two bits of the third octet are one. The low six bits are the code described below.

A TCP implementation MUST ignore this option (with the two upper bits in the third octet being one) when it is found in any other packet but the TCP packet sent as response to the first packet.

No TCP connection has been established between client and server when the server answered with this packet.

Typically the "V6REQ" option will be the only option in the TCP header in this case because more options make no sense in this case.

4.4. Reserved combination

The combination "zero-one" for the highest two bits in the third octet is reserved for future use:

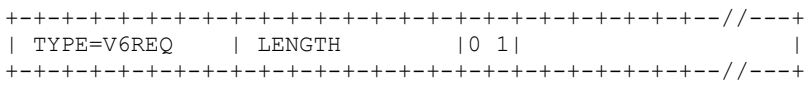

Figure 8: Reserved option format

A TCP implementation MUST ignore this option (with the two upper bits in the third octet being zero and one) when it is found in any TCP packet.

5. Option codes

5.1. List of option codes

The following option codes are defined:

Type*	Code	Length	Code
00	1	3	Strong IPv6 request
00	2	3	Weak IPv6 request
10	3	3	Supported for this port
10	4	3	Not supported for this port
10	5	4	Request not supported
11	6	21	Address to be used

(*) 00 = valid in request
10 = valid in an informational response
11 = valid in a force restart response

Figure 9: List of option codes

5.2. Strong IPv6 request

A client sends this option in the initial TCP packet whenever it desires to use IPv6 for the connection as far as it is possible.

If the server understands this option but there is no TCP/IPv6 port which is equivalent to the TCP/IPv4 port the client wants to connect to it MUST answer with "not supported for this port".

If the server understands this option AND there is a TCP/IPv6 port which is equivalent to the TCP/IPv4 port the client wants to connect to the server MUST answer with "address to be used".

5.3. Weak IPv6 request

This option is similar to the "strong IPv6 request" however it is used if the client does not necessarily prefer IPv6 over IPv4.

Mis en forme : Surlignage

Commentaire [Med12]: Any technical reasons for such preference?

If the server understands the option and there is a TCP/IPv6 equivalent for the TCP/IPv4 port but the server also does not prefer using IPv6 over IPv4 the server MUST answer with "supported for this port".

The client may later send a "strong IPv6 request" to get information about the TCP/IPv6 port that can be used instead of the TCP/IPv4 port.

In all other cases the server MUST react on this option the same way as it reacts on a "strong IPv6 request". Especially in the case that the server wants to communicate with the client over IPv6 instead of IPv4 it will answer with "address to be used".

5.4. Supported for this port

A server sends this option in its first packet (in the answer to the first packet of the TCP connection) as a response to a "weak IPv6 request".

Using this option it indicates that it would support a "strong IPv6 request" for this TCP/IPv4 port.

5.5. Not supported for this port

A server sends this option in its first packet as a response to a "strong" or "weak IPv6 request".

Using this option it indicates that the it is not possible to establish the desired TCP connection via IPv6.

5.6. Request not supported

A server sends this option in its first packet as a response to any "request" option that was not understood by the server.

This is the case when the initial packet from the client contained a "V6REQ" option whose topmost two bits of the third octet were zero but the server did not understand the meaning of that option. (For example if the "CODE" field did not have a value understood by the server.)

The server indicates that it ignored the option it did not understand.

Note that the "request not supported" option is only of informational kind; the client SHOULD NOT treat this as error message.

If the server did not understand the value of the "CODE" field the "request not supported" option has the following form:

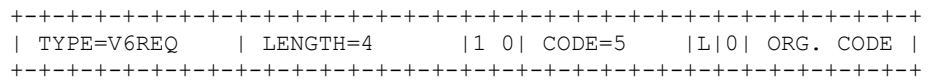

Figure 10: Request not supported option format

"ORG. CODE" is the value of the "CODE" field in the request not understood by the server.

"L" is zero if the code is not understood by the server at all. "L" is one if the server understands the code but the length of the option is not correct (currently: 3).

Note: If there will be "requests" (type="00") in the future being longer than three octets a "response" indicating "invalid arguments" should be defined. "Request not supported" should not be used in this case.

5.7. Address to be used

A server sends this option in a "force restart response" (Section 4.3) described above.

Doing so the server instructs the client to use IPv6 instead of IPv4 to connect to the server.

The option has the following form:

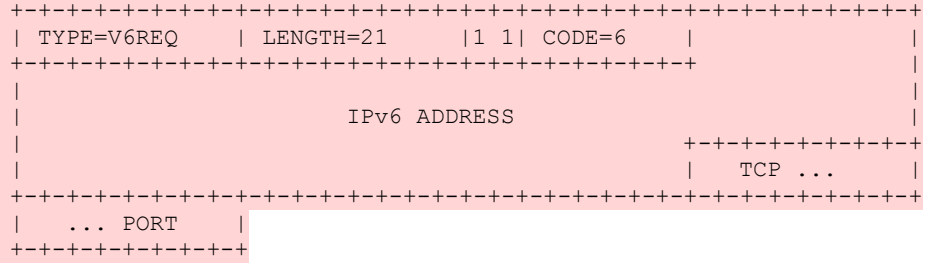

Figure 11: Address to be used option format

The "IPv6 ADDRESS" and "TCP PORT" fields are the IPv6 address and the TCP/IPv6 port number of a TCP port which is equivalent to the port the client wanted to connect to using IPv4.

Commentaire [Med13]: A misbehaving node can insert this address to force the traffic to be redirected to an illegitimate node. How to ensure that the address is not injected by an attacker?

Note that especially in the case of NAT46 the IPv6 address may be different for different TCP ports. The TCP port number to be used with IPv6 may also differ from the TCP port number to be used with IPv4 - especially in a NAT46 scenario.

The TCP/IPv6 port returned MUST be absolutely equivalent to the TCP/IPv4 port the client wanted to connect to. This means that when establishing a connection from an IPv6 client to the server there must be no difference in functionality when directly connecting to the TCP/IPv6 port returned by the server compared to connecting to the TCP/IPv4 port via a NAT64.

6. Using the option with IPv6

As described above the option may also used in TCP/IPv6 packets. This typically only makes sense if the server is a NAT.

If the server understands this option, it is a NAT and the client can connect to the actual server (behind the NAT) directly the NAT will respond to a "strong IPv6 request" with an "address to be used" answer.

If the server understands this option but it is not a NAT or the client cannot connect to the actual server directly it will respond with "not supported for this port".

A NAT may behave differently for different ports - e.g. if one port is forwarded to a server which can be accessed directly and another port is forwarded to a server which cannot be accessed directly.

The same is true when a "weak IPv6 request" is received; in this case the NAT may decide if it answers with a "supported for this port" option or with "address to be used".

7. Suggestion for IPv4 addresses assigned by the RIRs

IPv4 addresses are very rare.

Unfortunately there still seem to be ISPs who receive IPv4 addresses from the RIRs using these addresses for IPv4-only nodes.

To force these ISPs to use the IPv4 addresses they receive from the RIRs for dual-stack nodes and translation mechanisms only the author of the document has the following suggestion:

- RIRs are only allowed to assign IPv4 addresses if the receiver guarantees that the receiver will observe some rules for ALL IPv4 addresses used by the receiver.

- "ALL IPv4 addresses" means:

  - An organisation already having IPv4 addresses may only receive new IPv4 addresses from a RIR if it already observes the rules for all IPv4 addresses in use by this organisation. As soon as it receives the new IPv4 addresses observing the rules will be mandatory for the "old" IPv4 addresses!

  - In a situation where an organisation using IPv4 addresses being subject to these rules and one organisation using IPv4 addresses not being subject to these rules merge (e.g. in a company takeover) the merged organisation MUST decide if ALL IPv4 used by the organisation will be subject to these rules or if the IPv4 addresses being subject to these rules before the merge are returned to the RIR.

- The RIR should revoke the IPv4 addresses when the rules are violated.

Possible rules could be (note that most of these rules are already mandatory for "new implementations" according to RFC 6540 [RFC6540]):

- All functionality a node supports via IPv4 MUST also be supported via IPv6 - regardless if the functionality uses TCP, UDP or other protocols.

- This also applies to the DNS entry: There MUST be an "AAAA" record for each host name having an "A" record using such an address.

- The only exception is the use of a secondary host name for forcing an IPv4 connection like "www.ipv4.example.com" (force IPv4) instead of "www.example.com" (use the default protocol); the operator of the host MUST ensure that the secondary host name is only used in case of problems with the IPv6 connection. Especially it is not allowed to publish statements like: "The address of our home page is www.ipv4.example.com".

- On the other hand such a node MAY support functionality which is only accessible via IPv6 but not via IPv4.

- Nodes using these IPv4 addresses MUST prefer IPv6 over IPv4 (Example: If a node wants to establish a connection to a host having both an "AAAA" and an "A" DNS record these nodes MUST first try to use the "AAAA" record; only if this fails they MAY use the "A" entry.)

- All clients using these IPv4 addresses MUST use the "strong IPv6 request" option in outgoing TCP connections over IPv4. This will force the clients to use the direct IPv6 connection in the case of a NAT46. It will also guarantee that it is verifiable if the client using this IPv4 address supports IPv6.

- All servers using these IPv4 addresses MUST respond to a "strong" and/or "weak IPv6 request" using "address to be used" (and refuse a TCP connection via IPv4). This will reduce the load of NAT64 servers. It will also guarantee that it is verifiable if the server using this IPv4 address supports IPv6.

8. References

8.1. Normative References

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <https://www.rfc-editor.org/info/rfc793>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

8.2. Informational References

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <https://www.rfc-editor.org/info/rfc6146>.

[RFC6540] George, W., Donley, C., Liljenstolpe, C., and L. Howard, "IPv6 Support Required for All IP-Capable Nodes", BCP 177, RFC 6540, DOI 10.17487/RFC6540, April 2012, <https://www.rfc-editor.org/info/rfc6540>.

Author's Address

Martin D. J. Rosenau

Email: martin@rosenau-ka.de