

**Discovery Mechanism for QUIC-based, Non-transparent Proxy Services**  
**draft-kuehlewind-quic-proxy-discovery-00**

Abstract

Often an intermediate instance (such as a proxy server) is used to connect to a web server or a communicating peer if a direct end-to-end IP connectivity is not possible or the proxy can provide a support service like, e.g., address anonymisation. To use a non-transparent proxy a client explicitly connects to it and requests forwarding to the final target server. The client either knows the proxy address as preconfigured in the application or can dynamically learn about available proxy services. This document describes different discovery mechanisms for non-transparent proxies that are either located in the local network, e.g. home or enterprise network, in the access network, or somewhere else on the Internet usually close to the target server or even in the same network as the target server.

This document assumes that the non-transparent proxy server is connected via QUIC and discusses potential discovery mechanisms for such a QUIC-based, non-transparent proxy.

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 May 7, 2020.

## Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                       |   |                    |
|-----------------------|---|--------------------|
| <a href="#">1.</a>    | Introduction . . . . .                                      | <a href="#">2</a>  |
| <a href="#">2.</a>    | Using DHCP for Local Discovery . . . . .                    | <a href="#">3</a>  |
| <a href="#">3.</a>    | Using IPv6 Neighbor Discovery for Local Discovery . . . . . | <a href="#">5</a>  |
| <a href="#">3.1.</a>  | Using PVDs . . . . .  | <a href="#">6</a>  |
| <a href="#">4.</a>    | DNS Service Discovery (DNS-SD) . . . . .                    | <a href="#">7</a>  |
| <a href="#">4.1.</a>  | Local discovery using mDNS . . . . .                        | <a href="#">7</a>  |
| <a href="#">4.2.</a>  | Discovery for Remote Domains . . . . .                      | <a href="#">8</a>  |
| <a href="#">5.</a>    | Using PCP options . . . . .                                 | <a href="#">8</a>  |
| <a href="#">6.</a>    | Using Anycast address . . . . .                             | <a href="#">9</a>  |
| <a href="#">7.</a>    | IANA Considerations . . . . .                               | <a href="#">9</a>  |
| <a href="#">8.</a>    | Security Consideration . . . . .                            | <a href="#">10</a> |
| <a href="#">9.</a>    | Contributors . . . . .                                      | <a href="#">10</a> |
| <a href="#">10.</a>   | Acknowledgments . . . . .                                   | <a href="#">10</a> |
| <a href="#">11.</a>   | References . . . . .  | <a href="#">10</a> |
| <a href="#">11.1.</a> | Normative References . . . . .                              | <a href="#">10</a> |
| <a href="#">11.2.</a> | Informative References . . . . .                            | <a href="#">11</a> |
|                       | Authors' Addresses . . . . .                                | <a href="#">11</a> |

**[1.](#) Introduction**

QUIC is a new transport protocol that was initially developed as a way to optimize HTTP traffic by supporting multiplexing without head-of-line-blocking and integrating security directly into the transport. This tight integration of security allows the transport and security handshakes to be combined into a single round-trip exchange, after which both the transport connection and authenticated encryption keys are ready.

Often an intermediate instance (such as a proxy server) is used to connect to a web server or a communicating peer if a direct end-to-end IP connectivity is not possible or the proxy can provide a



support service like, e.g., address anonymisation. QUIC's ability to multiplex, encrypt data, and migrate between network paths makes it ideal for solutions that need to tunnel or proxy traffic.

Existing proxies that are based on TCP and HTTP are often transparent. That is, they do not require the cooperation of the ultimate connection endpoints, and are often not visible to one or both of the endpoints. If QUIC provides the basis for future tunneling and proxying solutions, it is expected that this relationship will change. At least one of the endpoints will be aware of the proxy, explicitly connect to it, and coordinate with it. This makes the proxy and tunneling non-transparent to at least most often the client. This allows client hosts to make explicit decisions about the services they request from proxies (for example, simple forwarding or more advance performance-optimizing services), and to do so using a secure communication channel between itself and the proxy. [[I-D.kuehlewind-quic-substrate](#)] describes some of the use cases for using QUIC for proxying and tunneling.

To use a non-transparent proxy service, a client explicitly connects to it and requests forwarding to the final target server. The client either knows the proxy address as preconfigured in the application or can dynamically learn about available proxy servers. This document describes different discovery mechanisms for proxies that are either located in the local network, e.g. home or enterprise network, in the access network, or somewhere else on the Internet usually close to the target server or even in the same network as the target server. For the rest of the document the work "proxy" refers to a non-transparent proxy.

The discovery mechanisms proposed in this document cover a range of approaches based on IETF protocols and commonly used mechanisms, however, other mechanisms in more specialized networks are possible as well. For 5G networks, the 3GPP specifies an extended exposure framework that potentially can also be used for proxy discovery and routing support.

## **2. Using DHCP for Local Discovery**

DHCP [[RFC2131](#)] can be used to announce the IP address of local proxy server in IPv4 networks, as well DHCPv6 [[RFC8415](#)] in IPv6 networks. New options for both protocols are specified below. The option can contain one or more IP addresses of QUIC-based proxy servers. All of the addresses share the same Lifetime value. If it is desirable to have different Lifetime values, multiple options can be used.



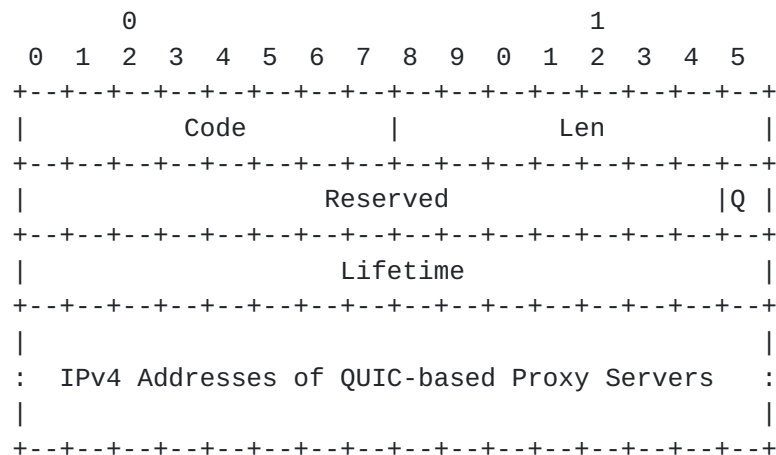


Figure 1: IPv4 Proxy Discovery DHCP option format

Code: Proxy Discovery option code (TDB) (8 bit)

Len: length of the option (without the Code and Len fields) in units of octets. The minimum value is 6 if one IPv4 address is contained in the option. Every additional IPv4 address increases the length by 4. (8-bit unsigned integer)

Q: is set to one if proxy supports QUIC on port 443 (1 bit)

Lifetime: maximum time in seconds (relative to the time the packet is received) over which these IP4 addresses can be used for proxy discovery. A value of all one bits (0xffff) represents infinity. A value of zero means that the proxy addresses SHOULD no longer be used. (16-bit unsigned integer)

IPv4 Addresses of QUIC-based Proxy Servers: one or more 64-bit IPv4 addresses of QUIC-based proxy servers. The number of addresses is determined by the Length field. That is, the number of addresses is equal to  $(\text{Length} - 2) / 4$ .



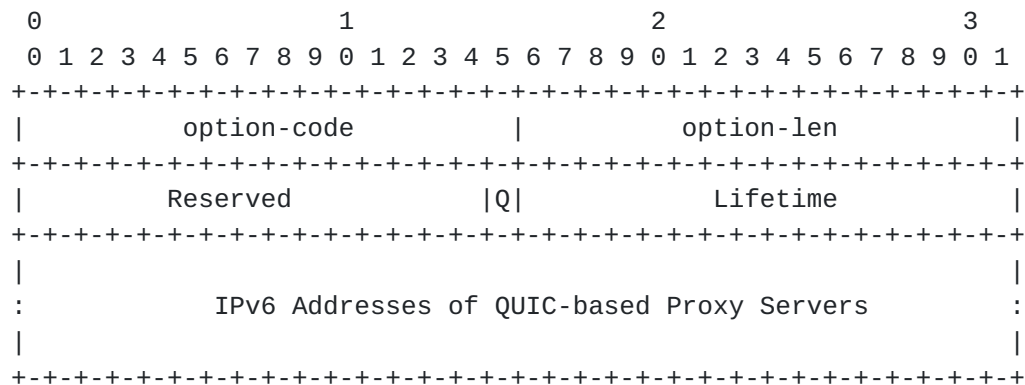


Figure 2: IPv4 Proxy Discovery DHCP option format

option-code: Proxy Discovery option code (TDB) (16 bit)

option-len: length of the option (without the Type and Length fields) in units of 8 octets. The minimum value is 20 if one IPv6 address is contained in the option. Every additional IPv6 address increases the length by 16. (8-bit unsigned integer)

Q: is set to one if proxy supports QUIC on port 443 (1 bit)

Lifetime: maximum time in seconds (relative to the time the packet is received) over which these IPv6 addresses can be used for proxy discovery. A value of all one bits (0xffffffff) represents infinity. A value of zero means that the proxy addresses SHOULD no longer be used. (16-bit unsigned integer)

IPv6 Addresses of QUIC-based Proxy Servers: one or more 128-bit IPv6 addresses of QUIC-based proxy servers. The number of addresses is determined by the Length field. That is, the number of addresses is equal to (Length - 1) / 2.

### 3. Using IPv6 Neighbor Discovery for Local Discovery

If a proxy is located in the local network, information to discover a proxy service can be provided in a new Router Advertisement (RA) Option [[RFC4861](#)], the Proxy Discovery option.





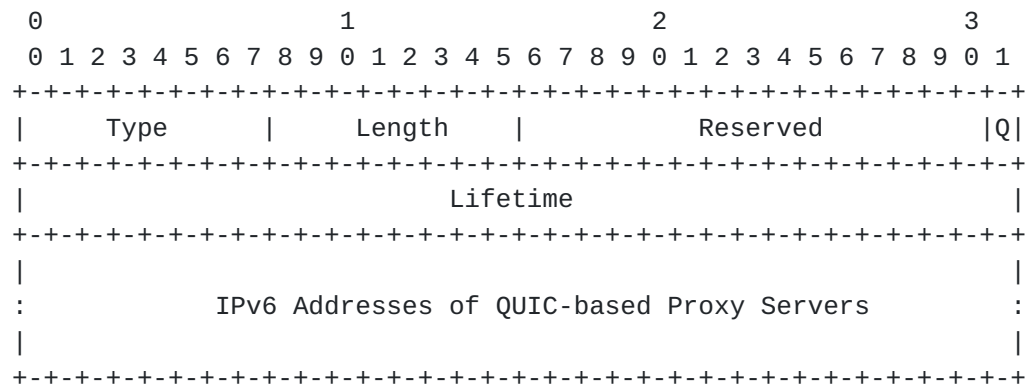


Figure 3: Proxy Discovery RA option format

Type: Proxy Discovery option type (TDB) (8 bit)

Length: length of the option (including the Type and Length fields) in units of 8 octets. The minimum value is 3 if one IPv6 address is contained in the option. Every additional IPv6 address increases the length by 2. (8-bit unsigned integer)

Q: is set to one if proxy supports QUIC on port 443 (1 bit)

Lifetime: maximum time in seconds (relative to the time the packet is received) over which these IPv6 addresses can be used for proxy discovery. A value of all one bits (0xffffffff) represents infinity. A value of zero means that the proxy addresses SHOULD no longer be used. (32-bit unsigned integer)

IPv6 Addresses of QUIC-based Proxy Servers: one or more 128-bit IPv6 addresses of QUIC-based proxy servers. The number of addresses is determined by the Length field. That is, the number of addresses is equal to (Length - 1) / 2.

### 3.1. Using PVDs

If the local network provides configuration with an Explicit Provisioning Domain (PvD) [[I-D.ietf-intarea-provisioning-domains](#)], the RA defined above can be used with the PvD Option or alternatively proxy information can be retrieved in the additional information JSON files associated with the PvD ID. The endhost resolves the URL provided in the PvD ID into an IP address using the local DNS server that is associated with the corresponding PvD (see also [section 3.4.4](#). [[I-D.ietf-intarea-provisioning-domains](#)]). If a QUIC-based proxy services is provided the additional information JSON file contains the key "QuicProxyIP". It can then optionally also contain more information about the specific proxy services offered using the "ProxyService" key. Or the client can connect directly to the proxy



over QUIC on port 443 and request information about the proxy service directly from the proxy server.

For remote network a Web PvD might be available that contains proxy information. If provided, the PvD JSON configuration file retrievable at the URI with the format:

```
https://<Domain>/.well-known/pvd"
```

#### **4. DNS Service Discovery (DNS-SD)**

[RFC6763] describes the use of SRV records to discover the available instances of a type of service. To get a list of names of the available instance for a certain service a client requests records of type "PTR" (pointer from one name to another in the DNS namespace [RFC1035] for a name containing the service and domain.

As specified in [RFC6763] the client can perform a PTR query for a list of available proxy instance in following way:

```
_quicproxy._udp.<domain>
```

here the <domain> portion is the domain name where the service is registered. The domain name can be obtained via DHCP options or preconfigured.

The result of this PTR lookup is a set of zero or more PTR records giving Service Instance names. Then to contact a particular service, the client can query for the SRV [RFC2782] and TXT records of the selected service instance name. The SRV record contains the IP address of the proxy service instance as well as the port number. The port number of QUIC-based proxy is usually expected to be 443 but may differ. The TXT can contain additional information describing the kind of proxy services that is offered.

##### **4.1. Local discovery using mDNS**

[RFC6762] defines the use of ".local." for performing DNS like operations on the local link. Any DNS query for a name ending "local." will be send to predefined IPv4 or IPv6 link local multicast address.

To discovery QUIC-based proxy services locally, the client request the PTR record for the name:

```
_quicproxy._udp.local.
```



The result of this PTR lookup is a set of zero or more PTR records giving Service Instance Names of the form:

```
<Instance>._quicproxy._udp.local.
```

Editors' Note: Or `_masque._udp` ? Or `_proxy._quic._udp` or `_quicproxy._http._udp` ...? However in the later case the proxy should probably also actually offer a webpage...

## 4.2. Discovery for Remote Domains

If a client wants to discover a QUIC-based proxy server for a remote domain, this domain has to be known by the client, e.g. being preconfigured in the application.

## 5. Using PCP options

Port Control Protocol (PCP), described in [RFC6887], defines mechanism to do packet forwarding for different types of IPv4/IPv6 Network Address Translators (NAT) or firewall. The usual deployment on PCP include Carrier-Grade NAT (CGN), Customer Premises Equipment (CPE) and as well as residential NATs. Hence, the discovery of QUIC-based proxy can also be realized via PCP implementations.

PCP allows options to be included in the PCP request and response header. The QUIC-based proxy information can be included in the response header as options. As [RFC6887] describes, the client will ignore any options that it does not understand.

A PCP option with QUIC-based proxy information is specified below.

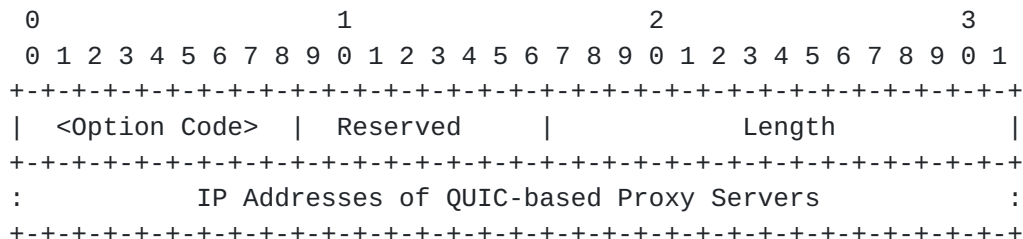


Figure 4: Proxy Discovery PCP option format

The fields are described below -

Option Code: 8 bits. The most significant bit indicates if this option is mandatory (0) or optional (1) to process.

Reserved: 8 bits. MUST be set to 0 on transmission and MUST be ignored on reception.



**Option Length:**

:16 bits. Indicates the length of the enclosed data, in octets. Options with length of 0 are allowed. Options that are not a multiple of 4 octets long are followed by one, two, or three 0 octets to pad their effective length in the packet to be a multiple of 4 octets. The Option Length reflects the semantic length of the option, not including any padding octets.

IP Addresses of QUIC-based Proxy Servers: one or more 128-bit IPv6 addresses and/or 32-bit IPv4 addresses of QUIC-based proxy servers. The number of addresses is determined by the Length field. That is, the number of addresses is equal to  $(\text{Length} - 1) / 2$ .

**6. Using Anycast address**

Wellknown IP anycast addresses can be used to start communicating with QUIC proxy or to discovery any or a list of unicast address of a QUIC proxy. When the proxy receives the request for proxy functionalities, it can either decide to respond to the client with the anycast address as source address or it can send back a list of unicast address with a redirect command.

TODO: complete the description

**7. IANA Considerations**

IANA is requested to assign two DHCP options, one for IPv4 and one for IPv6, in the "BOOTP Vendor Extensions and DHCP Options" registry (<http://www.iana.org/assignments/bootp-dhcp-parameters>), as specified in [RFC2939], and the "Option Codes" registry under DHCPv6 parameters (<http://www.iana.org/assignments/dhcpv6-parameters>), respectively, as well a new value for the Proxy Discovery Option in the IPv6 Neighbor Discovery Option Formats registry.

This document adds a key to the "Additional Information PvD Keys" registry, defined by [[I-D.ietf-intarea-provisioning-domains](#)].

| JSON key     | Description                           | Type                | Example                               |
|--------------|---------------------------------------|---------------------|---------------------------------------|
| -----        | -----                                 | -----               | ---                                   |
| QuicProxyIP  | IP address for<br>QUIC-based proxies  | Array of<br>Strings | "["2001:db8:::1",<br>"2001:db8:::2"]" |
| -----        | -----                                 | -----               | -----                                 |
| ProxyService | IDs identifying<br>a specific service | Array of<br>Strings | "["Forwarding",<br>"DNSResolution"]"  |
| -----        | -----                                 | -----               | -----                                 |





Further, IANA is requested to register a new service name "quicproxy" in the "Service Name and Transport Protocol Port Number Registry" (<https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>).

## **8. Security Consideration**

TBD

## **9. Contributors**

## **10. Acknowledgments**

## **11. References**

### **11.1. Normative References**

- [I-D.ietf-intarea-provisioning-domains]  
Pfister, P., Vyncke, E., Pauly, T., Schinazi, D., and W. Shao, "Discovering Provisioning Domain Names and Data", [draft-ietf-intarea-provisioning-domains-08](#) (work in progress), October 2019.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), DOI 10.17487/RFC2131, March 1997, <<https://www.rfc-editor.org/info/rfc2131>>.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", [RFC 2782](#), DOI 10.17487/RFC2782, February 2000, <<https://www.rfc-editor.org/info/rfc2782>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", [RFC 4861](#), DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", [RFC 6762](#), DOI 10.17487/RFC6762, February 2013, <<https://www.rfc-editor.org/info/rfc6762>>.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", [RFC 6763](#), DOI 10.17487/RFC6763, February 2013, <<https://www.rfc-editor.org/info/rfc6763>>.



- [RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", [RFC 6887](#), DOI 10.17487/RFC6887, April 2013, <<https://www.rfc-editor.org/info/rfc6887>>.
- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 8415](#), DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.

## **11.2. Informative References**

- [I-D.kuehlewind-quic-substrate]  
Kuehlewind, M., Sarker, Z., Fossati, T., and L. Pardue,  
"Use Cases and Requirements for QUIC as a Substrate",  
[draft-kuehlewind-quic-substrate-01](#) (work in progress),  
July 2019.
- [RFC2939] Droms, R., "Procedures and IANA Guidelines for Definition of New DHCP Options and Message Types", [BCP 43](#), [RFC 2939](#), DOI 10.17487/RFC2939, September 2000, <<https://www.rfc-editor.org/info/rfc2939>>.

### Authors' Addresses

Mirja Kuehlewind  
Ericsson

Email: [mirja.kuehlewind@ericsson.com](mailto:mirja.kuehlewind@ericsson.com)

Zaheduzzaman Sarker  
Ericsson

Email: [zaheduzzaman.sarker@ericsson.com](mailto:zaheduzzaman.sarker@ericsson.com)

