

**Simple Datagram Usability and Connectivity Kata
draft-pardue-quic-siduck-00**

Abstract

This document describes a simple application protocol for testing implementations of the QUIC DATAGRAM frame. SiDUCK (Simple Datagram Usability and Connectivity Kata) defines a new ALPN ID, "siduck-00", along with a basic offer and acknowledgement interaction using datagram payload data.

Note to Readers

Discussion of this document takes place on the QUIC Working Group mailing list (quic@ietf.org), which is archived at <https://mailarchive.ietf.org/arch/browse/quic/> [1].

Source for this draft and an issue tracker can be found at <https://github.com/lpardue/draft-pardue-quic-siduck> [2].

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 21, 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

1.	Introduction	2
1.1.	Notational Conventions	3
2.	Negotiating SiDUCK	3
3.	Protocol Behaviour	4
4.	The SiDUCK_ONLY_QUACKS_ECHO error code	4
5.	Known Issues / Problems	4
6.	Security Considerations	4
7.	IANA Considerations	4
8.	References	4
8.1.	Normative References	4
8.2.	Informative References	5
8.3.	URIs	5
Appendix A.	Acknowledgements	6
	Author's Address	6

[1.](#) Introduction

The DATAGRAM frame [[I-D.pauly-quic-datagram](#)] for QUIC [[I-D.ietf-quic-transport](#)] is an extension for the unreliable delivery of datagrams. Applications using DATAGRAM negotiate it using Transport Parameters but the application-specific contents are left for a higher-level mechanism.

HTTP/3 DATAGRAM [[I-D.schinazi-quic-h3-datagram](#)] defines the usage of QUIC DATAGRAM frames when the application protocol running over QUIC is HTTP/3 [[I-D.ietf-quic-http](#)], which is negotiated using ALPN [[RFC7301](#)] with an identifier such as "h3-24". The HTTP/3 DATAGRAM frame is transmitted as the QUIC DATAGRAM frame Datagram Data field. It contains a flow identifier field that can be used for demultiplexing applications, and an HTTP/3 Datagram Payload field whose semantics are defined by individual applications.

Since the QUIC DATAGRAM and HTTP/3 DATAGRAM specifications leave the semantic higher-order behaviour undefined, it is difficult to prove the interoperability of implementations. This document defines SiDUCK (Simple Datagram Usability and Connectivity Kata), a new

Pardue

Expires May 21, 2020

[Page 2]

application that maps directly to QUIC DATAGRAM and uses the Datagram Data field directly to support a simple client-server echo service.

Using QUIC DATAGRAM directly provides the benefit of avoiding complexity of HTTP/3 session establishment and flow identifier allocation and demultiplexing. SiDUCK's simpler application semantics allow implementers to exercise the interoperability of foundational layers of DATAGRAM, building progressive assurance that benefits the design and robustness of other DATAGRAM-using applications.

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

The terms sh-token and sh-boolean are imported from [[STRUCTURED-HEADERS](#)].

Example HTTP requests and responses use the HTTP/2-style formatting from [[RFC7540](#)].

This document uses the variable-length integer encoding from [[I-D.ietf-quic-transport](#)].

2. Negotiating SiDUCK

SiDUCK is an application protocol that runs atop QUIC. The token "siduck" is used to identify SiDUCK in ALPN, used during QUIC connection establishment. Only implementations of the final, published RFC can identify themselves as "siduck". Until such an RFC exists, implementations MUST NOT identify themselves using this string.

RFC Editor's Note: Please remove the following prior to publication of a final version of this document.

Implementations of draft versions of the protocol MUST add the string "-" and the corresponding draft number to the identifier. For example, [draft-pardue-quic-siduck-00](#) is identified using the string "siduck-00".

Non-compatible experiments that are based on these draft versions MUST append the string "-" and an experiment name to the identifier. For example, an experimental implementation based on [draft-pardue-quic-siduck-09](#) which extends the permitted message types to include "honk" might identify itself as "siduck-09-goose". Note that any

label MUST conform to the "token" syntax defined in [Section 3.2.6 of \[RFC7230\]](#).

3. Protocol Behaviour

A SiDUCK session consists of a client and server. Once a SiDUCK session is established, the client SHOULD send a QUIC DATAGRAM frame with the Datagram Data field containing "quack". A server that receives a quack, MUST respond with a QUIC DATAGRAM frame with the Datagram Data field containing "quack-ack".

A client MUST NOT send any other message than "quack". A server that receives a non-quack message MUST close the connection with an error of type `SIDUCK_ONLY_QUACKS_ECHO`.

4. The `SIDUCK_ONLY_QUACKS_ECHO` error code

This document defines the `SIDUCK_ONLY_QUACKS_ECHO` QUIC Application error code. It has the value `0x101`.

5. Known Issues / Problems

- o The design does not permit large DATAGRAM frames. Sending these in a simple application protocol would be beneficial and within the scope of SiDUCK's goals.
- o The design does not specify requirements on Transport Parameters. For example, since streams are not used, the spec could require that endpoints send `max_streams_{bidi|uni}` with the value 0.

6. Security Considerations

There are not believed to be any further security considerations beyond those presented in [\[I-D.ietf-quic-transport\]](#).

7. IANA Considerations

TBD

8. References

8.1. Normative References

[I-D.ietf-quic-transport]

Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", [draft-ietf-quic-transport-23](#) (work in progress), September 2019.

[I-D.pauly-quic-datagram]

Pauly, T., Kinnear, E., and D. Schinazi, "An Unreliable Datagram Extension to QUIC", [draft-pauly-quic-datagram-04](#) (work in progress), October 2019.

[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>>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", [RFC 7230](#), DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.

[RFC7301] Friedl, S., Popov, A., Langley, A., and E. Stephan, "Transport Layer Security (TLS) Application-Layer Protocol Negotiation Extension", [RFC 7301](#), DOI 10.17487/RFC7301, July 2014, <<https://www.rfc-editor.org/info/rfc7301>>.

[STRUCTURED-HEADERS]

Nottingham, M. and P. Kamp, "Structured Headers for HTTP", [draft-ietf-httpbis-header-structure-14](#) (work in progress), October 2019.

8.2. Informative References

[I-D.ietf-quic-http]

Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", [draft-ietf-quic-http-23](#) (work in progress), September 2019.

[I-D.schinazi-quic-h3-datagram]

Schinazi, D., "Using QUIC Datagrams with HTTP/3", [draft-schinazi-quic-h3-datagram-01](#) (work in progress), October 2019.

[RFC7540] Belshé, M., Peon, R., and M. Thomson, Ed., "Hypertext Transfer Protocol Version 2 (HTTP/2)", [RFC 7540](#), DOI 10.17487/RFC7540, May 2015, <<https://www.rfc-editor.org/info/rfc7540>>.

8.3. URIs

[1] <https://mailarchive.ietf.org/arch/browse/quic/>

[2] <https://github.com/lpardue/draft-pardue-quic-siduck>

[Appendix A](#). Acknowledgements

Thanks to Andy Stitzer who achieved interop of an unpublished early SiDUCk specification during the IETF 106 Hackathon.

Author's Address

Lucas Pardue

Email: lucaspardue.24.7@gmail.com