Network Working Group
Request for Comments: 4993
Category: Standards Track

A. Newton
VeriSign, Inc.

August 2007

A Lightweight UDP Transfer Protocol for the Internet Registry Information Service

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document describes a lightweight UDP transfer protocol for the Internet Registry Information Service (IRIS). This transfer protocol uses a single packet for every request and response, and optionally employs compression over the contents of the packet.

Table of Contents

1. Introduction

Using Straightforward Name Authority Pointers (S-NAPTR) [4], IRIS has the ability to define the use of multiple application transports or transfer protocols for different types of registry services, all at the discretion of the server operator. The UDP transfer protocol defined in this document is completely independent of the registry types for which it can carry data.

The binding of this UDP transfer protocol to IRIS is called IRIS-LWZ (for IRIS Lightweight using Compression). Its message exchange pattern is simple: a client sends a request in one UDP packet, and a server responds with an answer in one UDP packet.

IRIS-LWZ packets are composed of two parts, a binary payload descriptor and a request/response transaction payload. The request/response transaction payload may be compressed using the DEFLATE [1] algorithm.

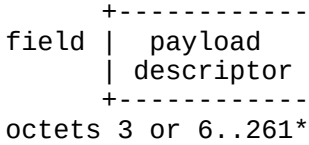
2. Document Terminology

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 RFC 2119 [6].

Octet fields with numeric values are given according to the conventions in RFC 1166 [10]: the leftmost bit of the whole field is the most significant bit; when a multi-octet quantity is transmitted the most significant octet is transmitted first. Bits signifying flags in an octet are numbered according to the conventions of RFC 1166 [10]: bit 0 is the most significant bit and bit 7 is the least significant bit. When a diagram describes a group of octets, the order of transmission for the octets starts from the left.

3. Packet Format

The packet format for IRIS-LWZ is as follows:



* In request packets, the payload descriptor can vary in length from 6 to 261 octets (i.e., 6..261). In response packets, the payload descriptor is always 3 octets.

IRIS-LWZ Packet

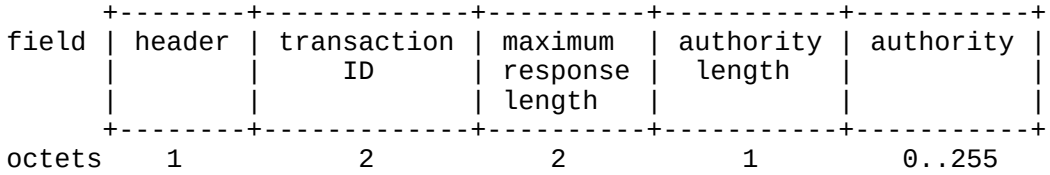
Each IRIS-LWZ query or response is contained in a single UDP packet. Servers MUST be prepared to accept packets as large as 4000 octets, and clients MUST NOT send packets larger than 4000 octets.

3.1. Payload Descriptor

The payload descriptor has two different formats, one for a request and one for a response. However, each format shares a common 1-octet payload header described in Section 3.1.3.

3.1.1. Payload Request Descriptor

The payload descriptor for request packets varies from 6 to 261 octets in length and has the following format:



Request Payload Descriptor

These fields have the following meanings:

o header - as described in Section 3.1.3.

o transaction ID - a 16-bit value identifying the transaction. This value will be returned in the payload response descriptor (Section 3.1.2) and can be used by clients to match requests with

responses. Clients SHOULD NOT use sequential values (see Section 8). Clients MUST NOT set all the bits in this value to 1 (i.e., use a value of 0xFFFF).

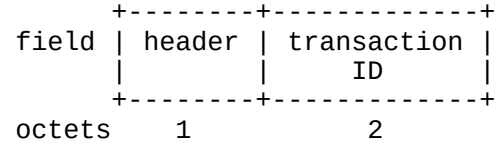
- o maximum response length - the total length of the UDP packet (i.e., UDP header length + payload descriptor length + XML payload length) that should not be exceeded when responding to this request. If the server cannot provide a response that is equal to or less than this value, then it MUST respond with size information (Section 3.1.6).

- o authority length - the length of the authority field in this payload descriptor.

- o authority - a string of octets describing the authority against which this request is to be executed. See [3] for the definition and description of an authority. The number of octets in this string MUST be no more and no less than the number specified by the authority length.

3.1.2. Payload Response Descriptor

The payload descriptor for response packets is always 3 octets and consists of a payload header (Section 3.1.3) and a transaction ID.



Payload Response Descriptor

The purpose of the transaction ID is to allow clients to match requests to responses. A value of 0xFFFF is reserved for server use. The value of the transaction ID is as follows:

1. If the transaction ID in the corresponding request could not be read due to truncation, servers MUST use a transaction ID with all bits set to 1 (i.e., a value of OxFFFF) and send a descriptor error (see Section 3.1.7).

2. If the transaction ID in the corresponding request is a value of 0xFFFF, servers MUST use a transaction ID of 0xFFFF and send a descriptor error (see Section 3.1.7).

3. Otherwise, the transaction ID MUST be the value of the transaction ID of the corresponding request.

3.1.3. Payload Header

The bits of the payload header are ordered according to RFC 1166 [10], where bit 0 is the most significant and bit 7 is the least significant. Each bit in the 1-octet payload header has the following meaning:

- bits 0 and 1 - version number ('V' field) - If 0 (both bits are zero), the protocol is the version defined in this document. Otherwise, the rest of the bits in the header and the payload may be interpreted as another version.

- bit 2 - request/response flag ('RR' flag) - If 0, this packet is a request (Section 3.1.1) packet. If 1, this packet is a response (Section 3.1.2) packet.

- bits 3 - payload deflated ('PD' flag) - If 1, the payload is compressed using the DEFLATE [1] algorithm.

- bit 4 - deflate supported ('DS' flag) - If 1, the sender of this packet supports compression using the DEFLATE algorithm. When this bit is 0 in a request, the payload of the response MUST NOT be compressed with DEFLATE.

- bit 5 - reserved - This MUST be 0.

- bits 6 and 7 - The value of these bits indicates payload types (Section 3.1.4) ('PT' field).

3.1.4. Payload Types

A payload type indicates the type of content in the UDP packet following the payload descriptor. Some payload types have no meaning in request packets, and some payload types differ in meaning between requests and responses. Some payload types indicate an empty payload.

The payload type values in binary are as follows:

00 - xml payload ('xml' type). The payload is either an IRIS-based XML request or an IRIS-based XML response.

01 - version info ('vi' type). In a request packet, this payload type indicates that the server is to respond with version information (Section 3.1.5), and that the payload is empty. In a response packet, this payload type indicates that the payload is version information (Section 3.1.5).

10 - size info ('si' type). This payload type has no meaning in a request packet and is a descriptor error. In a response packet, this payload type indicates that the payload is size information (Section 3.1.6).

11 - other info ('oi' type). This payload type has no meaning in a request packet and is a descriptor error. In a response packet, this payload type indicates that the payload is other information (Section 3.1.7).

3.1.5. Version Information

A payload type with version information ('vi') MUST be conformant to the XML defined in [8] and use the <versions> element as the root element.

In the context of IRIS-LWZ, the protocol identifiers for these elements are as follows:

<transferProtocol> - the value "iris.lwz1" to indicate the protocol specified in this document.

<application> - the XML namespace identifier for IRIS [3].

<dataModel> - the XML namespace identifier for IRIS registries.

This document defines no extension identifiers and no authentication mechanism identifiers.

Servers SHOULD send version information in the following cases:

1. In response to a version information request (i.e., the PT field is set to 'vi').

2. The version in a payload descriptor header does not match a version the server supports.

3. The IRIS-based XML payload does not match a version the server supports.

The protocols identified by the <transferProtocol> element MUST only indicate protocols running on the same socket as the sender of the corresponding response. In other words, while a server operator may also be running IRIS-XPC [9], this XML instance is only intended to describe version negotiation for IRIS-LWZ.

The octet size for the 'requestSizeOctets' and 'responseSizeOctets' attributes of the <tranferProtocol> element are defined in Section 3.1.6.

3.1.6. Size Information

A payload type with size information ('si') MUST be conformant to the XML defined in [8] and use the <size> element as the root element.

Octet counts provided by this information are defined as the total length of the UDP packet (i.e., UDP header length + payload descriptor length + XML payload length).

3.1.7. Other Information

A payload type with other information ('oi') MUST be conformant to the XML defined in [8] and use the <other> element as the root element.

The values for the 'type' attribute of <other> are as follows:

'descriptor-error' - indicates there was an error decoding the descriptor. Servers SHOULD send a descriptor error in the following cases:

1. When a request is received with a payload type indicating size information (i.e., the PT field is 'si').

2. When a request is received with a payload type indicating other information (i.e., the PT field is 'oi').

3. When a request is sent with a transaction ID of 0xFFFF (which is reserved for server use).

4. When a request is received with an incomplete or truncated payload descriptor.

5. When reserved bits in the payload descriptor are set to values other than zero.

'payload-error' - indicates there was an error interpreting the payload. Servers MUST send a payload error if they receive XML (i.e., the PT field is set to 'xml') and the XML cannot be parsed.

'system-error' - indicates that the receiver cannot process the request due to a condition not related to this protocol. Servers SHOULD send a system-error when they are capable of responding to requests but not capable of processing requests.

'authority-error' - indicates that the intended authority specified in the corresponding request is not served by the receiver. Servers SHOULD send an authority error when they receive a request directed to an authority other than those they serve.

'no-inflation-support-error' - indicates that the receiver does not support payloads that have been compressed with DEFLATE [1]. Servers MUST send this error when they receive a request that has been compressed with DEFLATE but they do not support inflation.

4. Interactions

The intent of IRIS-LWZ is to utilize UDP for IRIS requests and responses when UDP is appropriate. Not all IRIS requests and responses will be able to utilize UDP and may require the use of other transfer protocols (i.e., IRIS-XPC [9] and/or Blocks Extensible Exchange Protocol (BEEP)). The following strategy SHOULD be used:

1. If a request requires authentication, confidentiality, or other security, use another transfer protocol. IRIS-XPC [9] is RECOMMENDED.

2. The maximum packet size should be calculated as follows:

   a. If the path MTU is unknown, the maximum packet size MUST be 1500 octets.

   b. If the path MTU is known, the maximum packet size MUST NOT exceed the path MTU and MUST NOT exceed 4000 octets.

3. If a request is less than or equal to the maximum packet size, send it uncompressed.

4. If a request can be compressed to a size less than or equal to the maximum packet size, send the request using compression. Otherwise, use another transfer protocol. In cases where another transfer protocol is needed, IRIS-XPC [9] is RECOMMENDED.

5. If a request yields a size error, send the request with another transfer protocol. IRIS-XPC [9] is RECOMMENDED.

For retransmission of requests considered to be unanswered, a client SHOULD retransmit using a timeout value initially set to 1 second. This timeout value SHOULD be doubled for every retransmission, and a client SHOULD NOT retransmit any request once the timeout value has reached 60 seconds.

Clients that use timeout values other than the recommendations above MUST allocate or have allocated dedicated network resources that will ensure fairness to other network packets and avoid network congestion.

Clients MUST NOT have more than one outstanding request (i.e., an unanswered request that has not timed out) at a time unless they allocate or have been allocated dedicated network bandwidth and resources reserved specifically for this purpose.

Finally, if a client intends multiple requests to the same server in a short amount of time, this protocol offers no real advantage over IRIS-XPC [9]. In such a case, IRIS-XPC is RECOMMENDED to be used as it would be similarly or more efficient and would offer greater response sizes and allow better security.

5. Internationalization Considerations

XML processors are obliged to recognize both UTF-8 and UTF-16 [2] encodings. Use of the XML defined by [8] MUST NOT use any other character encodings other than UTF-8 or UTF-16.

6. IRIS Transport Mapping Definitions

This section lists the definitions required by IRIS [3] for transport mappings.

6.1. URI Scheme

See Section 7.1.1.

6.2. Application Protocol Label

See Section 7.1.3.

7. IANA Considerations

7.1. Registrations

7.1.1. URI Scheme Registration

URL scheme name: iris.lwz

Status: permanent

URL scheme syntax: defined in [3].

Character encoding considerations: as defined in RFC 3986 [5].

Intended usage: identifies an IRIS entity made available using XML over UDP

Applications using this scheme: defined in IRIS [3].

Interoperability considerations: n/a

Security Considerations: defined in Section 8.

Relevant Publications: IRIS [3].

Contact Information: Andrew Newton <andy@hxr.us>

Author/Change controller: the IESG

7.1.2. Well-known UDP Port Registration

Protocol Number: UDP

UDP Port Number: 715

Message Formats, Types, Opcodes, and Sequences: defined in Sections 3 and 3.1.

Functions: defined in IRIS [3].

Use of Broadcast/Multicast: none

Proposed Name: IRIS-LWZ

Short name: iris.lwz

Contact Information: Andrew Newton <andy@hxr.us>

7.1.3. S-NAPTR Registration

Application Protocol Label (see [4]): iris.lwz

Intended usage: identifies an IRIS server using XML over UDP

Interoperability considerations: n/a

Security Considerations: defined in Section 8.

Relevant Publications: IRIS [3].

Contact Information: Andrew Newton <andy@hxr.us>

Author/Change controller: the IESG

8. Security Considerations

IRIS-LWZ is intended for serving public data; it provides no in-band mechanisms for authentication or confidentiality. Any application with these needs must provide out-of-band mechanisms (e.g., IPsec), or use the IRIS transfer protocols that provide such capabilities, such as IRIS-XPC [9].

Due to this lack of security, it is possible for an attacker to alter IRIS-LWZ messages sent from the client to the server and from the server to the client. Such an attack can result in denying usage of an IRIS service or in supplying false information to end users and many other scenarios.

Because IRIS-LWZ is a UDP-based protocol, it is possible for servers using IRIS-LWZ to be used in a type of distributed denial-of-service attack known as a reflection attack. This type of attack affects other types of UDP-using protocols, such as DNS. Server operators should be prepared to apply the same methods used for mitigating reflection attacks with other protocols, such as DNS, when using IRIS-LWZ. All operators should follow the advice given in BCP 38 [7].

IRIS-LWZ uses transaction IDs in the payload descriptors to better enable a client to match a response to a request. By randomizing the transaction IDs being used (i.e., not using sequential numbers), attackers flooding the network with a large amount of spoofed packets have a lesser chance of succeeding with the attack. This measure is not guaranteed to thwart any such attack. Client implementers MUST take appropriate measures when ignoring this advice.

9. References

9.1. Normative References

[1] Deutsch, P., "DEFLATE Compressed Data Format Specification version 1.3", RFC 1951, May 1996.

[2] The Unicode Consortium, "The Unicode Standard, Version 3", ISBN 0-201-61633-5, 2000, <The Unicode Standard, Version 3>.

[3] Newton, A. and M. Sanz, "IRIS: The Internet Registry Information Service (IRIS) Core Protocol", RFC 3981, January 2005.

[4] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

[5] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[6] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCP 14, March 1997.

[7] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[8] Newton, A., "A Common Schema for Internet Registry Information Service Transfer Protocols", RFC 4991, August 2007.

[9] Newton, A., "XML Pipelining with Chunks for the Internet Registry Information Service", RFC 4992, August 2007.

9.2. Informative References

[10] Kirkpatrick, S., Stahl, M., and M. Recker, "Internet numbers", RFC 1166, July 1990.

Appendix A. Examples

This section gives examples of IRIS-LWZ exchanges. Lines beginning with "C:" denote data sent by the client to the server, and lines beginning with "S:" denote data sent by the server to the client. Following the "C:" or "S:", the line contains either octet values in hexadecimal notation with comments or XML fragments. No line contains both octet values with comments and XML fragments. Comments are contained within parentheses.

The following example demonstrates an IRIS client requesting a lookup of 'AUP' in the 'local' entity class of a 'dreg1' registry. The client passes a bag (see [3]) with the search request. The server responds with a 'nameNotFound' response and an explanation.

```
C:           (request packet)
C: 0x08      (header: V=0,RR=request,PD=no,DS=yes,PT=xml)
C: 0x03 0xA4 (transaction ID=932)
C: 0x05 0xDA (maximum response size=1498)
C: 0x09      (authority length=9)
C:           (authority="localhost")
C: 0x6c 0x6f 0x63 0x61 0x6c 0x68 0x6f 0x73 0x74
C:           (IRIS XML request)
C: <request xmlns="urn:ietf:params:xml:ns:iris1"
C:    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
C:    <searchSet>
C:      <bag>
C:        <simpleBag xmlns="http://example.com/">
C:          <salt>127.0.0.1:3434</salt>
C:          <md5>4LnQ1KdCahzyvwBqJis5rw==</md5>
C:        </simpleBag>
C:      </bag>
C:      <lookupEntity
C:        registryType="dreg1"
C:        entityClass="local"
C:        entityName="AUP" />
C:    </searchSet>
C: </request>

S:           (response packet)
S: 0x20      (header: V=0,RR=response,PD=no,DS=no,PT=xml)
S: 0x03 0xA4 (transaction ID=932)
S:           (IRIS XML response)
S: <iris:response xmlns:iris="urn:ietf:params:xml:ns:iris1">
S: <iris:resultSet><iris:answer></iris:answer>
S: <iris:nameNotFound><iris:explanation language="en-US">
S: The name 'AUP' is not found in 'local'.</iris:explanation>
S: </iris:nameNotFound></iris:resultSet></iris:response>
```

Example 1

The following example demonstrates an IRIS client requesting domain availability information for 'milo.example.com'. The server responds that the domain is assigned and active.

```
C:           (request packet)
C: 0x00      (header: V=0,RR=request,PD=no,DS=no,PT=xml)
C: 0x0B 0xE7 (transaction ID=3047)
C: 0x0F 0xA0 (maximum response size=4000)
C: 0x0B      (authority length=11)
C:           (authority="example.com")
C: 0x65 0x78 0x61 0x6D 0x70 0x6C 0x65 0x23 0x63 0x6F 0x6D
C:           (IRIS XML request)
C: <request xmlns="urn:ietf:params:xml:ns:iris1"
C:   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:   xsi:schemaLocation="urn:ietf:params:xml:ns:iris1 iris.xsd" >
C:   <searchSet>
C:     <lookupEntity
C:       registryType="urn:ietf:params:xml:ns:dchk1"
C:       entityClass="domain-name"
C:       entityName="milo.example.com" />
C:   </searchSet>
C: </request>

S:           (response packet)
S: 0x20      (header: V=0,RR=response,PD=no,DS=no,PT=xml)
S: 0x0B 0xE7 (transaction ID=3047)
S:           (IRIS XML response)
S: <iris:response xmlns:iris="urn:ietf:params:xml:ns:iris1">
S: <iris:resultSet><iris:answer><domain
S: authority="example.com" registryType="dchk1"
S: entityClass="domain-name" entityName="tcs-com-1"
S: temporaryReference="true"
S: xmlns="urn:ietf:params:xml:ns:dchk1"><domainName>
S: milo.example.com</domainName><status><assignedAndActive/>
S: </status></domain></iris:answer>
S: </iris:resultSet></iris:response>
```

Example 2

The following example demonstrates an IRIS client requesting domain availability information for felix.example.net, hobbes.example.net, and daffy.example.net. The client does not support responses compressed with DEFLATE, and the maximum UDP packet it can safely receive is 498 octets. The server responds with size information indicating that it would take 1211 octets to provide an answer.

```
C:           (request packet)
C: 0x00      (header: V=0,RR=request,PD=no,DS=no,PT=xml)
C: 0x7E 0x8A (transaction ID=32394)
C: 0x01 0xF2 (maximum response size=498)
C: 0x0B      (authority length=11)
C:           (authority="example.net")
C: 0x65 0x78 0x61 0x6D 0x70 0x6C 0x65 0x23 0x6E 0x65 0x74
C:           (IRIS XML request)
C: <request xmlns="urn:ietf:params:xml:ns:iris1"
C:   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:   xsi:schemaLocation="urn:ietf:params:xml:ns:iris1 iris1.xsd">
C:   <searchSet>
C:     <lookupEntity registryType="dchk1" entityClass="domain-name"
C:       entityName="felix.example.net" />
C:   </searchSet>
C:   <searchSet>
C:     <lookupEntity registryType="dchk1" entityClass="domain-name"
C:       entityName="hobbes.example.net" />
C:   </searchSet>
C:   <searchSet>
C:     <lookupEntity registryType="dchk1" entityClass="domain-name"
C:       entityName="daffy.example.net" />
C:   </searchSet>
C: </request>

S:           (response packet)
S: 0x22      (header: V=0,RR=response,PD=no,DS=no,PT=si)
S: 0x7E 0x8A (transaction ID=32394)
S:           (Size Information XML response)
S: <responseSize xmlns="urn:ietf:params:xml:ns:iris-transport">
S:   <octets>1211</octets>
S: </responseSize>
```

Example 3

The following example illustrates an IRIS client requesting the version information from a server, and the server returning the version information.

```
C:           (request packet)
C: 0x01      (header: V=0,RR=request,PD=no,DS=no,PT=vi)
C: 0x2E 0x9C (transaction ID=11932)
C: 0x01 0xF2 (maximum response size=498)
C: 0x0B      (authority length=11)
C:           (authority="example.net")
C: 0x65 0x78 0x61 0x6D 0x70 0x6C 0x65 0x23 0x6E 0x65 0x74

S:           (response packet)
S: 0x21      (header: V=0,RR=response,PD=no,DS=no,PT=vi)
S: 0x2E 0x9C (transaction ID=11932)
S:           (Version Information XML response)
S: <versions xmlns="urn:ietf:params:xml:ns:iris-transport">
S:   <transferProtocol protocolId="iris.lwz1">
S:     <application protocolId="urn:ietf:params:xml:ns:iris1">
S:       <dataModel protocolId="urn:ietf:params:xml:ns:dchk1"/>
S:       <dataModel protocolId="urn:ietf:params:xml:ns:dreg1"/>
S:     </application>
S:   </transferProtocol>
S: </versions>
```

Example 4

Appendix B. Contributors

Substantive contributions to this document have been provided by the members of the IETF's CRISP Working Group, especially Milena Caires and David Blacka.

Author's Address

Andrew L. Newton
VeriSign, Inc.
21345 Ridgetop Circle
Sterling, VA 20166
USA

Phone: +1 703 948 3382
EMail: andy@hxr.us
URI: http://www.verisignlabs.com/

Full Copyright Statement

Intellectual Property

Acknowledgement

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