Network Working Group
Internet-Draft
Expires: October 31, 2005

A. Newton
VeriSign, Inc.

April 29, 2005

A Lightweight UDP Transfer Protocol for the the Internet Registry Information Service
draft-ietf-crisp-iris-lwz-02

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on October 31, 2005.

Copyright Notice

Abstract

This document describes a lightweight UDP transfer protocol for the Internet Registry Information Service (IRIS).

Table of Contents

1. Introduction

Using 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 descretion of the server operator. The UDP transfer protocol defined in this document is completely modular and may be used by any registry types.

The binding of this UDP transfer protocol to IRIS is called IRIS-LWZ (for IRIS Lightweight using Compression).

IRIS-LWZ is composed of two parts, a binary payload descriptor and an 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 RFC2119 [6].

3. Packet Format

The UDP packet format for IRIS-LWZ is as follows:

field	src port	dest port	checksum	UDP length	payload descriptor	payload
octets	2	2	2	2	1..261	0..n

(where "src port" means source port and "dest port" means destination port).

Each IRIS-LWZ query or response is contained in a single UDP packet.

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 has the following format:

field	header	transaction ID	maximum response length	authority length	authority
octets	1	2	2	1	0..255

These fields have the following meanings:

header - as described in Section 3.1.3.

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 pick the value randomly and SHOULD NOT use sequences of 16 bit values. Clients MUST NOT set all the bits in this value to 1 (i.e. use a value of 0xFFFF).

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

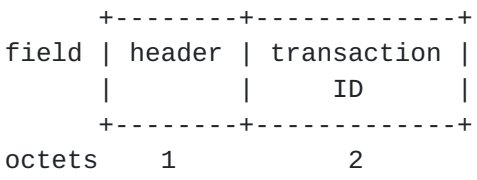
information (Section 3.1.3.1.2).

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

authority - a string of octets describing the authority against wich 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 consists of a payload header (Section 3.1.3) and a transaction ID.



The transaction ID MUST be the value of the transaction ID of the corresponding request. If the corresponding request did not contain a transaction ID, servers MUST use a transaction ID with all bits set to 1 (i.e. use a value of 0xFFFF) and send a descriptor error (see Section 3.1.3.1.3).

3.1.3 Payload Header

Each bit in the 1 byte payload header has the following meaning:

bits 7 and 6 - version number ('V' flag) - 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 5 - 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 4 - payload deflated ('PD' flag) - If 1, the payload is compressed using the DEFLATE [1] algorithm.

bit 3 - 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 2 - reserved - This MUST be 0.

bits 1 and 0 - The value of these bits indicate payload types (Section 3.1.3.1) ('PT' flag).

3.1.3.1 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 3.1.3.1.1), and that the payload is empty. In a response packet, this payload type indicates that the payload is version information (Section 3.1.3.1.1).

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 3.1.3.1.2).

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 3.1.3.1.3).

3.1.3.1.1 Version Information

A payload type with version information ('vi') MUST be comformant to the XML defined in [7] 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 flag 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 request. In other words, while a server operator may also be running IRIS-XPC, this XML instance is only intended to describe version negotiation for IRIS-LWZ.

3.1.3.1.2 Size Information

A payload type with size information ('si') MUST be comformant to the XML defined in [7] and use the <responseSize> element as the root element.

3.1.3.1.3 Other Information

A payload type with other information ('oi') MUST be comformant to the XML defined in [7] 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 flag is 'si').

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

3. When a request is sent with an invalid transaction ID.

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

'payload-error' - indicates there was an error interpretting the payload. Servers MUST send a payload error if they receive XML (i.e. the PT flag 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. Internationalization Considerations

XML processors are obliged to recognize both UTF-8 and UTF-16 [2] encodings. Use of the XML defined by [7] MUST NOT use any other character encodings other than UTF-8 or UTF-16.

5. IRIS Transport Mapping Definitions

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

5.1 URI Scheme

See Section 6.1.1.

5.2 Application Protocol Label

See Section 6.1.3.

6. IANA Considerations

6.1 Registrations

6.1.1 URI Scheme Registration

URL scheme name: iris.lwz

URL scheme syntax: defined in Section 5.1 and [3].

Character encoding considerations: as defined in RFC2396 [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 7.

Relevant Publications: IRIS [3].

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

Author/Change controller: the IESG

6.1.2 Well-known UDP Port Registration

Protocol Number: UDP

Message Formats, Types, Opcodes, and Sequences: defined in Section 3 and Section 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>

6.1.3 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 7.

Relevant Publications: IRIS [3].

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

Author/Change controller: the IESG

7. Security Considerations

IRIS-LWZ is intended for serving public data; it provides no in-band mechanisms for authentication or encryption. Any application with this need must provide out of band mechanisms to provide it (e.g., IPSec), or use the IRIS transfer protocols that provides such capabilities.

8. 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, "Internet Registry Information Service", RFC 3891, January 2004.

[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 Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

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

[7] Newton, A., "A Common Schema for Internet Registry Information Service Transfer Protocols", draft-ietf-crips-iris-common-transport-00 (work in progress), April 2005.

Author's Address

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

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

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 either contains 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 parenthesis.

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 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>
```

Figure 4: 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>
```

Figure 5: 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>
```

Figure 6: Example 3

The following example illustrates an IRIS client requesting the version information from a server, and the server returning the verion 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.lwz">
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>
```

Figure 7: Example 4

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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