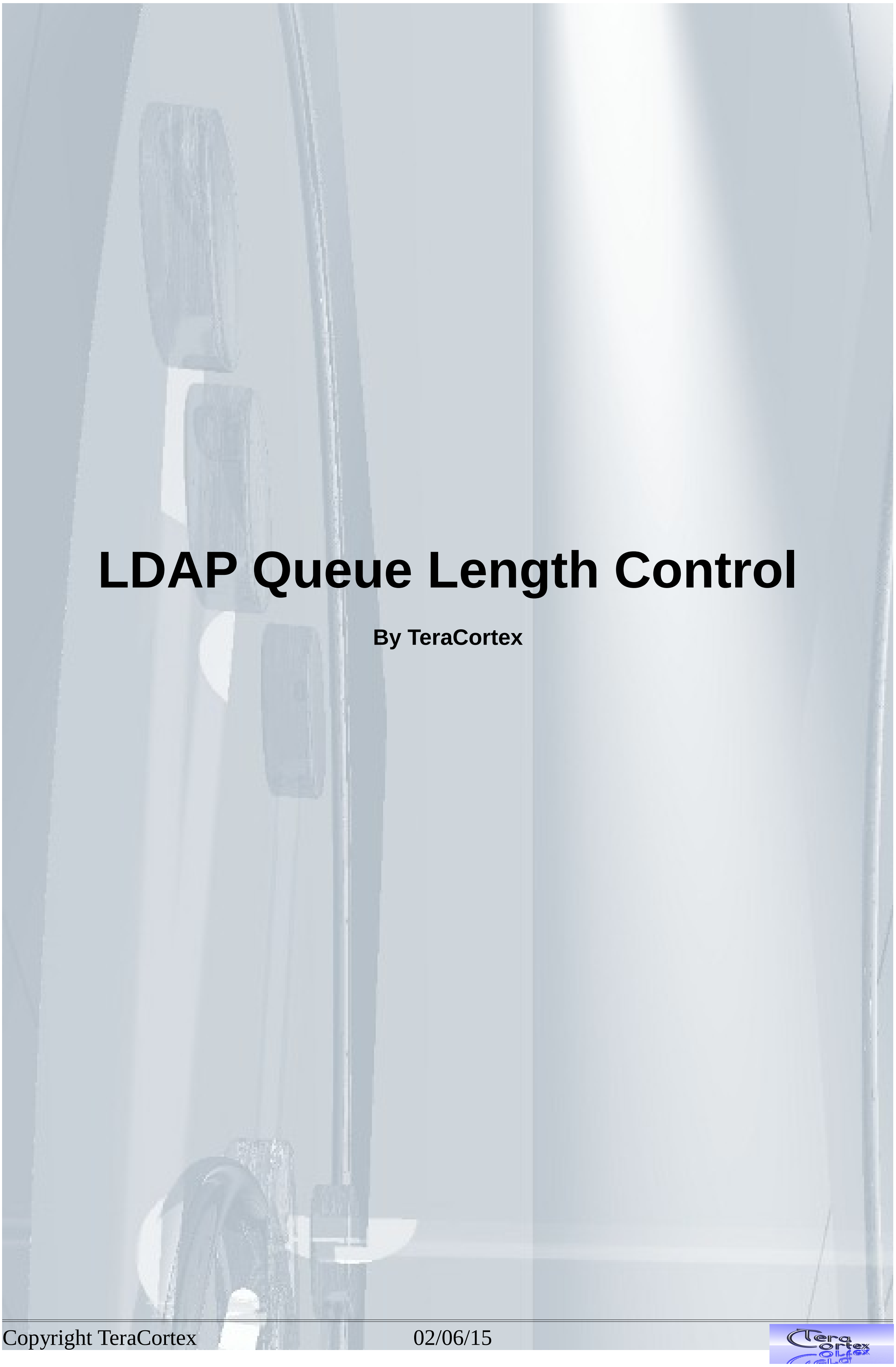

LDAP Queue Length Control

By TeraCortex

Table of Contents

Status of This Memo........2

Abstract........2

Copyright Notice........3

1. Overview........4

1.1. Conventions and Terminology........5

2. The Queue Length Control........6

2.1. Client Behavior........6

2.2. Server Behavior........7

3. Interaction with other Controls........8

3.1. Transaction Control........8

4. Handling of extended requests........8

5. Security Considerations........8

6. IANA Considerations........9

7. Acknowledgments........9

8. References........9

8.1. Normative References........9

8.2. Informative References........10

Author's address........10

Independent Submission
INTERNET-DRAFT
Intended Status: Proposed Standard
Expires 2015/08/26
draft-hollstein-queuelength-control-03.txt

Christian Hollstein
TeraCortex

February 2015

LDAP Queue Length Control

Status of This Memo

This document is not an Internet Standards Track specification. It is published for examination, experimental implementation, and evaluation. Distribution of this memo is unlimited.

Abstract

This document specifies a new control. The client can use it to tell the server the number of responses it expects for a series of requests that were sent in asynchronous mode. This enables the server to better pack usually small LDAP result messages into send buffers that better match the TCP maximum transfer unit. From this a substantial improvement in network ressource utilization, response time and throughput is expected.

Copyright Notice

1. Overview

POSIX - compatible TCP stacks use the Nagle Algorithm to achieve a good utilization of network packets. They try to fill a TCP packet completely before it is sent. On LDAP level this means that the client must wait for a response until the server side operating system decides to send the packet. Or the LDAP server employs TCP_NODELAY to let the system dump any message immediately on the wire, regardless of its size. This situation leads either to wait times on client side or to poor network utilization for LDAP response messages.

LDAP has no protocol element to let the client tell the server the number of responses it expects when the requests were sent in asynchronous mode. This forces the server to assume that the client works in synchronous mode. After having received a request the server alternatively could of course wait for the next one before it sends the response for the first request. But the client waits at the same time for the first response, so the second request will not appear at the server site. In this situation the server could only take a decision how to proceed when the receive operation times out. This mechanism ruins performance and is no real option.

With the queue length control the client can tell the server how many responses it expects. The server can refrain from sending responses until the given number of requests have been processed. Then it can pack all responses into a send buffer that will more closely match the TCP maximum transfer unit (MTU) of the operating system. With the exception of search results, LDAP response messages are usually very small (less than 20 bytes) while common operating system have MTUs around 1500 bytes. Still the server can set the TCP_NODELAY option on the socket because itself cares for optimum packet utilization. This optimizes both sides: Response time AND throughput.

1.1. Conventions and 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].

2. The Queue Length Control

The control object identifier is an IANA - assigned OID. The criticality is always FALSE. The control value is the requested queue length and takes the fowllowing form:

queue-length = INTEGER (1 .. N)

2.1. *Client Behavior*

Clients MAY include this control in the first of a sequence of requests sent in asynchronous mode. The total number of requests in the sequence MUST be equal to the given queue-length. Clients MUST NOT include this control in any other request of the same sequence. After having sent the whole sequence clients MUST wait for the number of responses given in queue-length. Clients MUST NOT use this control in any LDAP request type except ADD, DELETE, MODIFY, MODDN, COMPARE, SEARCH. Request of these types MAY be member of the same sequence without carrying the queue length control. An ABANDON request may be member of the same sequence but MUST NOT contain the queue length control. In the context of this specification result messages of a single SEARCH request are seen as just one response, regardless how many objects are in the SEARCH result set. The use of this control in EXTENDED requests depends on the precise semantics of the request.

2.2. Server Behavior

In addition to the standard control semantics given in [RFC4511] servers supporting this control MUST behave as follows:

- If the requested queue length exceeds server side limitations the server responds with adminLimitExceeded (11).

- If the requested queue length is less than one the server responds with protocolError (2).

- If the requested queue length is acceptable but the sequence of requests contains request types not allowed the server responds with protocolError (2). The allowed request types are defined in chapter 2.1.

- If the client sends one or more queue length controls before a previous one was completely processed, the server responds with protocolError (2).

- If the requested queue length is acceptable the server processes each request in the queue and stores the responses in a buffer. When the current sequence of requests have been processed, the servers sends the response buffer.

In the context of the queue length control the server MUST handle the entire collection of result messages to a particular SEARCH request as just one response, regardless how many objects are in the SEARCH result set.

3. Interaction with other Controls

3.1. Transaction Control

Clients MUST NOT use the queue length control in conjunction with the transaction control. [RFC5805] requires that servers send their responses immediately inside of ongoing transactions.

Servers seeing both controls present in a LDAP request MUST ignore the queue length control.

4. Handling of extended requests

The transaction begin extended request [RFC5805] is synchronous by its nature and MUST not contain the queue length control. The transaction end extended request [RFC5805] MUST not contain the queue length request. It MAY be member of an asynchronous queue started earlier. The relation of the queue length control to other extended requests is left unspecified.

5. Security Considerations

General security considerations [RFC4510], especially those associated with update operations [RFC4511], apply to this extension.

6. IANA Considerations

IANA is asked to assign a new object identifier for this control.

7. Acknowledgments

The author gratefully acknowledges the contributions made by Internet Engineering Task Force participants.

8. References

8.1. Normative References

[RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997.

[RFC4510]   Zeilenga, K., Ed., "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map", RFC 4510, June 2006.

[RFC4511]   Sermersheim, J., Ed., "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, June 2006.

8.2. Informative References

[RFC5805] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP) Transactions", RFC 5805, March 2010.

Author's address

Christian Hollstein
TeraCortex
Hopfenbrede 2
D-49179 Ostercappeln

E-Mail: chollstein@teracortex.com
Phone: 0049 / 5473 / 9933
Mobile: 0049 / 160 / 96220958