ALTO
Internet-Draft
Intended status: Standards Track
Expires: September 3, 2009

M. Stiemerling
S. Kiesel
NEC Europe Ltd.

March 2, 2009

ALTO H1/H2 Protocol
draft-stiemerling-alto-h1h2-protocol-00

Status of this Memo

Copyright Notice

Abstract

Many Internet applications are used to access resources, such as pieces of information or server processes, which are available in several equivalent replicas on different hosts. This includes, but is not limited to, peer-to-peer file sharing applications. The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource. This memo proposes one possible way of implementing the ALTO protocol, called H1H2. The H1H2 protocol is a client/server protocols between end hosts and ALTO servers that allows two different ways of exchanging data between the server and the client.

Table of Contents

1. Introduction

Many Internet applications are used to access resources, such as pieces of information or server processes, which are available in several equivalent replicas on different hosts. This includes, but is not limited to, peer-to-peer file sharing applications. The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource. This memo proposes one possible way of implementing the ALTO protocol, called H1H2. The H1H2 protocol is a client/server protocols between end hosts and ALTO servers.

The problem space of ALTO is described in [I-D.marocco-alto-problem-statement] and the set of requirements is discussed in [I-D.kiesel-alto-reqs].

Comments and discussions about this protocol proposal should be directed to the ALTO working group: alto@ietf.org.

2. Solution Space

The ALTO protocol is a client/server protocol, operating between a number of ALTO clients and an ALTO server, as sketched in Figure 1

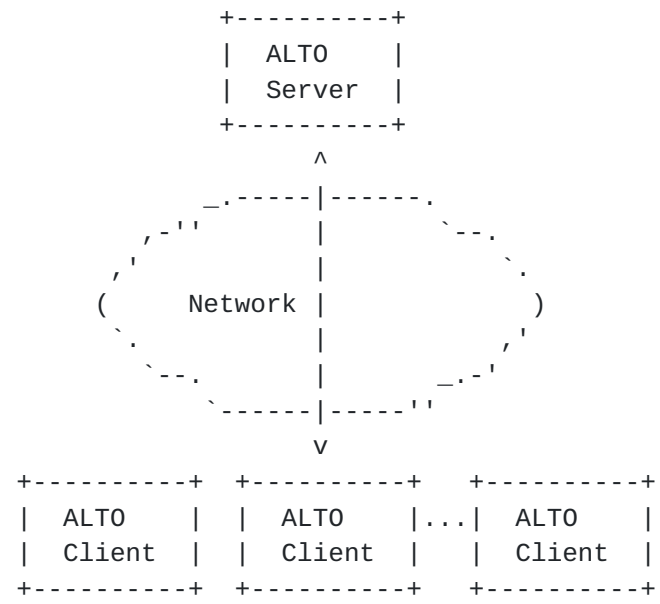

Figure 1: Network Overview of ALTO Protocol

An ALTO server stores information about preferences (e.g., a list of preferred autonomous systems, IP ranges, etc) and ALTO clients can retrieve these preferences. However, there are basically two different approaches on where the preferences are actually processed:

1. The ALTO server has a list of preferences and clients can retrieve this list via the ALTO protocol. This preference list can be partially updated by the server. The actual processing of the data is done on the client and thus there is no data of the client's operation revealed to the ALTO server . This approach has been proposed by [I-D.shalunov-alto-infoexport].

2. The ALTO server has a list of preferences or preferences calculated during runtime and the ALTO client is sending information of its operation (e.g., a list of IP addresses) to the server. The server is using this operational information to determine its preferences and returns these preferences (e.g., a sorted list of the IP addresses) back to the ALTO client. This approach has been initially described in [ACM.ispp2p], but never been described on the protocol level.

Approach 1 (we call it H1) has the advantage (seen from the client) that all operational information stays within the client and is not

revealed to the provider of the server. On the other hand, does approach 1 require that the provider of the ALTO server, i.e., the network operator, reveals information about its network structure (e.g., AS numbers, IP ranges, topology information in general) to the ALTO client.

Approach 2 (we call it H2) has the advantage (seen from the operator) that all operational information stays with the ALTO server and is not revealed to the ALTO client. On the other hand, does approach 2 require that the clients send their operational information to the server.

Both approaches have their pros and cons and are extensively discussed on the ALTO mailing list. But there is basically a dilemma: Approach 1 is seen as the only working solution by peer-to-peer software vendors and approach 2 is seen as the only working by the network operators. But neither the software vendors nor the operators seem to willing to change their position. However, there is the need to get both sides on board, to come to a solution.

Therefore, does this memo proposes to integrate both approaches in one protocol and offer a way for clients and servers to learn each preferred way of operating.

3. Proposed Solution

The current proposed solution is not yet defining a bit level syntax but describes the protocol on a high-level, i.e. it is more incomplete than complete.

The H1H2 protocol uses TCP as transport protocol between clients and server and some encoding of the messages to be defined later on.

The basic mechanism of the H1H2 protocol is to introduce an offer/ answer mechanism in the SETUP message of the protocol. The SETUP message is the first message sent from client to the server after the TCP session setup. The client includes is preference of data exchange, i.e., whether is willing to run H1, H2, or both. The server can then reply with its decisions, either to accept the clients choice, or why choice or to reject the choice.

Depending on the agreed mode, either H1 or H2, the protocol will proceed. The exact mechanism are TBD in future revisions.

However, this can lead to deadlock situations where clients ask only for H1 and the server insists on H2, i.e., there won't be an agreement between both ends. Meaning that there is no gain for both sides - the above described dilemma is basically still unsolved.

4. Security Considerations

This initial version of this memo does not yet have any security considerations, but they will be added in future revision.

5. Conclusion

This memo presents a very basic straw man protocol, is for sure work in progress, and is requesting feedback from the ALTO working group. Ask the authors why it is called H1H2 and not different.

6. References

6.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

6.2. Informative References

[ACM.ispp2p]
Aggarwal, V., Feldmann, A., and C. Scheideler, "Can ISPs and P2P systems co-operate for improved performance?", In ACM SIGCOMM Computer Communications Review (CCR), 37:3, pp. 29-40.

[I-D.kiesel-alto-reqs]
Kiesel, S., Popkin, L., Previdi, S., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", draft-kiesel-alto-reqs-01 (work in progress), November 2008.

[I-D.marocco-alto-problem-statement]
Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", draft-marocco-alto-problem-statement-04 (work in progress), February 2009.

[I-D.shalunov-alto-infoexport]
Shalunov, S., Penno, R., and R. Woundy, "ALTO Information Export Service", draft-shalunov-alto-infoexport-00 (work in progress), October 2008.

Authors' Addresses

Martin Stiemerling
NEC Laboratories Europe/University of Goettingen
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 113
Fax: +49 6221 4342 155
Email: stiemerling@nw.neclab.eu
URI: http://www.net.informatik.uni-goettingen.de/people/martin_stiemerling

Sebastian Kiesel
NEC Laboratories Europe
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 232
Fax: +49 6221 4342 155
Email: kiesel@nw.neclab.eu
URI: http://www.nw.neclab.eu/