Application-Layer Traffic Optimization (ALTO) Requirements
draft-kiesel-alto-reqs-01.txt

Sebastian Kiesel (Editor) <kiesel@nw.neclab.eu>
IETF-73, ALTO WG, Minneapolis, 2008-11-18

draft-kiesel-alto-reqs-01.txt

Authors

- S. Kiesel, Editor, NEC
- L. Popkin, Pando Networks, Inc.
- S. Previdi, Cisco Systems, Inc.
- R. Woundy, Comcast Corp.
- Y. R. Yang, Yale University

Further contributors

- V. Gurbani, Alcatel Lucent
- J. Livingood, Comcast Corp.
- E. Marocco, Telecom Italia
- S. Niccolini, NEC
- J. Seedorf, NEC
- M. Stiemerling, NEC

Related research projects

- NAPA-WINE (EC 7th FP) [NEC people]
- P4P working group [L. Popkin, Y.R. Yang]

Discussions before ALTO WG mailing list was established:

- http://ubiq.tilab.com/pipermail/alto-reqs/   (archives)

draft-kiesel-alto-reqs-01.txt

- External events since publication of -00 version
- Changes from draft-kiesel-alto-reqs-00 to -01
- Requirements overview
- Open issues & next steps

External events since publication of -00

- ALTO BoF at IETF #72 in Dublin + aftermath on p2pi mailing list
- Extensive discussions about focus and charter
  - Initial focus is P2P apps, finding "good" resource providers for <u>one</u> given resource consumer (i.e., "two-body-problem", as opposed to the n-body-problem of finding media relays between <u>two</u> end points)
  - …
- ALTO WG + charter approved. WG Deliverables acc. to charter:
  - Problem statement doc., including terminology
  - "A requirements document. This document will list requirements for the ALTO service, identifying, for example, types of information P2P applications may need for optimizing their choices."
  - A request/response protocol for querying the ALTO service […]
  - Defining core request and response formats and semantics
  - ALTO server discovery mechanism

Changes from draft-kiesel-alto-reqs-00 to -01

- Terminology aligned with, and terminology definition moved to draft-marocco-alto-problem-statement-03
  - In the future, we may have an own terminology draft as a working document

  - The problem statement draft currently defines:

Application,	Transport Address,	ALTO Service,
Overlay Network,	Host Location Attribute,	ALTO Server,
Peer,		ALTO Client,
Resource,	Local Traffic,	ALTO Query,
Resource Identifier,	Peering Traffic,	ALTO Reply,
Resource Provider,	Transit Traffic,	ALTO Transaction
Resource Consumer,		
Resource Directory,		

Changes from draft-kiesel-alto-reqs-00 to -01

- Minor changes, partly based on feedback to -00 version

    - More precise wording reflects achievable goals for ALTO (better-than-random-selection vs. optimal solution)
        - "optimize performance" ➔ "improve performance",
        - "minimize resource consumption" ➔ "reduce resource consumption"
        - etc.

    - Requirement level keywords
    - Lifetime allows soft decay of information instead of hard deadline

Changes from draft-kiesel-alto-reqs-00 to -01

- Remove the implicit assumption of a
  "sorting oracle"-style [Aggarwal, Feldmann] solution
  - Basically, removed Req #5 of the -00 draft
  - Also had to change some other requirements wrt. security

- TBD: section on overload control
  - It still assumes that every client issues ALTO queries "frequently"
  - Currently, does not explicitly name a transport protocol
    (in particular, not UDP)
  - Just "use TCP" may be enough, maybe not,
    e.g., should we just query another server if first one is overloaded?
  - Requirements for transport protocol and overload control
    is probably specific to overall solution

draft-kiesel-alto-reqs-01 – table of contents

- Introduction

- ALTO Requirements
  - ALTO Interface   Req. #1 - #6
  - Error handling and overload protection for ALTO   Req. #7 - #14
  - Security and Privacy   Req. #15 - #22

- Example rating criteria
  - Topological distance (e.g., AS hops)
  - Expected cost

Next steps (1)

- The authors are aware of these approaches:
  - Sorting Oracle [Aggarwal, Feldmann]
  - ALTO Information Export Service [Shalunov]
  - P4P [P4P Group]

- Currently, the requirements are very general
  - Fulfilled by all of these approaches
  - Few requirements are specific to one of them

- How can we get more specific?

Next steps (2)

- Goal: Definition of a "core set" of attributes
  for expressing preference (syntax + semantics)
  - The ALTO protocol will be extensible,
    i.e., one can define own attributes later
  - Nevertheless, we should come up with a "core set" of attributes
    that are useful in a broad variety of situations

- So far, the draft gives only two very vague examples

- Input from simulations and field tests needed!

- NB: ALTO is not only about improving the performance of the
  overlay, but also about reducing costs for operators!

draft-kiesel-alto-reqs-01.txt

Backup slides

Requirements: ALTO interface

- REQ. 1: The ALTO service MUST implement one or several well-defined interfaces, which can be queried from ALTO clients.

- REQ. 2: The ALTO clients MUST be able to query information from the ALTO service, which provides guidance for selecting appropriate resource providers.

- REQ. 3: ALTO clients MUST be able to find out where to send ALTO queries.

Requirements: ALTO interface

- REQ. 4: One mode of ALTO operation is that ALTO clients may be embedded directly in the resource consumer (e.g., peer of an unstructured P2P application), which wants to access a resource.

  However, another mode of operation is to perform ALTO queries indirectly, via resource directories. These translate a resource identifier to a list of resource providers with their corresponding transport addresses. The resource directories may issue ALTO queries to solicit preference on such lists, considering the respective resource consumer.

  The ALTO protocol MUST support both modes of operation. One way to fulfill this requirement is to include in the ALTO query a a host location attribute of the resource consumer, i.e., the entity that will eventually perform the data transfer.

Requirements: ALTO interface

- REQ. 5: The syntax and semantics of the ALTO protocol MUST be extensible to allow the requirements of future applications to be adopted. This includes, amongst others, support for adding protocol extensions in a non-disruptive, backward-compatible way, as well as protocol versioning support to clearly distinguish between incompatible major versions of the protocol.

- REQ. 6: The information available from the ALTO service is not a replacement for congestion control mechanisms. Applications using ALTO MUST ensure that they do not cause congestion in the network, e.g., by using TCP transport, which includes congestion control mechanisms.

Error handling and overload protection

- REQ. 7: Any application designed to use ALTO MUST also work reasonably if no ALTO servers can be found or if no responses to ALTO queries are received, e.g., due to connectivity problems or overload situation.

- REQ. 8: An overloaded ALTO server receiving too many requests MAY silently discard excess requests.

- REQ. 9: An ALTO client MAY retransmit an unanswered ALTO query after a reasonable amount of time, or it MAY query a different server. Otherwise, or if all retransmissions or queries to different servers have failed as well, the ALTO client MUST report to the application that no ALTO information is available. In this case, the application has to perform the resource provider selection without ALTO guidance.

Error handling and overload protection

- REQ. 10: An ALTO client MUST limit the total number of unanswered ALTO queries on a per-server basis. This limit MUST be reduced if a request times out and MAY be increased if several subsequent queries succeed without a timeout.

- REQ. 11: If an ALTO query cannot be sent because the maximum number of outstanding queries is reached, the ALTO client MAY wait for some time. Then, if it is still not possible to send the query, it MUST report to the application that no ALTO information is available. In this case, the application has to perform the resource provider selection without ALTO guidance.

Error handling and overload protection

- REQ. 12: An ALTO server, which is operating close to its capacity limit, SHOULD be able to inform clients about its impending overload situation, even if it has not yet to discard excess query messages. An ALTO client receiving a reply message with this overload indication may use the message content for the intended purpose (i.e., guidance with respect to resource provider selection). However, with respect to overload control, it MUST behave as if it had not received a reply.

Error handling and overload protection

- REQ. 13: The ALTO protocol MAY have a mechanism by which the ALTO client can specify the required level of precision. If only a medium amount of data has to be exchanged, it may be sufficient to get a quick answer from the ALTO service, which results in a certain improvement compared to a resource provider selection without any ALTO guidance. If, however, very large amounts of data will be transferred or the association will persist for an extended time, the client might request the ALTO service to spend more resources to make a recommendation that leads to higher improvements.

Error handling and overload protection

- REQ. 14: The ALTO protocol SHOULD support lifetime attributes, to enable caching of recommendations at ALTO clients. The ALTO protocol MAY specify an aging mechanism, which allows to give newer recommendations precedence over older ones.

Security and Privacy

- REQ. 15: The ALTO protocol MUST be designed in a way that the ALTO service can be provided by an operator which is not the operator of the IP access network.

- REQ. 16: Different instances of the ALTO service operated by different providers must be able to coexist.

- REQ. 17: The ALTO protocol MUST support mechanisms for mutual authentication and authorization of ALTO clients and servers.

Security and Privacy

- REQ. 18: The ALTO protocol MUST support different levels of detail in queries and responses, in order for the operator of an ALTO service to be able to control how much information (e.g., about the network topology) is disclosed.

- REQ. 19: The ALTO protocol MUST support different levels of detail in queries and responses, in order to protect the privacy of users, to ensure that the operators of ALTO servers and other users of the same application cannot derive sensitive information.

Security and Privacy

- REQ. 20: One ALTO interface SHOULD be defined in a way, that the operator of one ALTO server cannot easily deduce the resource identifier (e.g., file name in P2P file sharing) which the resource consumer seeking ALTO guidance wants to access.

- REQ. 21: If the ALTO protocol supports including privacy-sensitive information (such as resource identifiers or transport addresses) in the ALTO queries or replies, the protocol MUST also support encryption, in order to protect the privacy of users with respect to third parties.

- REQ. 22: The ALTO protocol MUST include appropriate mechanisms to protect the ALTO service against DoS attacks.

Example rating criteria

- Topological distance between the resource consumer and the candidate resource provider, e.g., the number of traversed autonomous systems (AS), or whether the traffic will be local traffic, peering traffic, or transit traffic.

- Expected cost for data exchange between the candidate resource provider and the resource consumer, according to the economic agreements between ISP. They may be expressed as absolute costs or relative costs, compared to retrieving the same data from another candidate resource provider.

Rating criteria that SHOULD NOT be used by the ALTO service include:

- Performance metrics related to instantaneous congestion status. This has to be probed and adapted to continuously, e.g., using TCP.