ALTO
Internet-Draft
Intended status: Informational
Expires: October 1, 2017

S. Kiesel
University of Stuttgart
M. Stiemerling
H-DA

March 30, 2017

Application Layer Traffic Optimization (ALTO) Cross-Domain Server Discovery
draft-ietf-alto-xdom-disc-00

Abstract

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications that have to select one or several hosts from a set of candidates capable of providing a desired resource. ALTO is realized by a client-server protocol. Before an ALTO client can ask for guidance it needs to discover one or more ALTO servers that can provide suitable guidance.

In some deployment scenarios, in particular if the information about the network topology is partitioned and distributed over several ALTO servers, it may be needed to discover an ALTO server outside of the own network domain, in order to get appropriate guidance. This document details applicable scenarios, itemizes requirements, and specifies a procedure for ALTO cross-domain server discovery.

Technically, the algorithm specified in this document takes one IP address and a U-NAPTR Service Parameter (i.e., "ALTO:http" or "ALTO:https") as parameters. It performs DNS lookups (for NAPTR resource records in the in-addr.arpa. or ip6.arpa. tree) and returns one or more URI(s) of information resources related to that IP address.

Terminology and Requirements Language

This document makes use of the ALTO terminology defined in RFC 5693 [RFC5693].

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 [RFC2119].

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications that have to select one or several hosts from a set of candidates capable of providing a desired resource [RFC5693]. ALTO is realized by an HTTP-based client-server protocol [RFC7285], which can be used in various deployment scenarios [I-D.ietf-alto-deployments].

1.1. Multiple Information Sources and Partitioned Knowledge

The ALTO base protocol document [RFC7285] specifies the communication between an ALTO client and a single ALTO server. It is implicitly assumed that this server can answer any query, possibly with some kind of default value if no exact data is known. No special provisions were made for the case that the ALTO information originates from multiple sources, which are possibly under the control of different administrative entities (e.g., different ISPs) or that the overall ALTO information is partitioned and stored on several ALTO servers.

1.1.1. Classification of Solution Approaches

Various protocol extensions and other solutions have been proposed to deal with multiple information sources and partitioned knowledge. They can be classified as follows:

1 Ensure that all ALTO servers have the same knowlegde

1.1 Ensure data replication and synchronization within the provisioning protocol (cf. RFC 5693, Fig 1 [RFC5693]).

1.2 Use an Inter-ALTO-server data replication protocol. Possibly, the ALTO protocol itself - maybe with some extensions - could be used for that purpose; however, this has not been studied in detail so far.

2 Accept that different ALTO servers (possibly operated by different organizations, e.g., ISPs) do not have the same knowledge

2.1 Allow ALTO clients to send arbitrary queries to any ALTO server (e.g. the one discovered using [RFC7286]). If this server cannot answer the query itself, it will fetch the data on behalf of the client, using the ALTO protocol or a to-be-defined inter-ALTO-server request forwarding protocol.

2.2 Allow ALTO clients to send arbitrary queries to any ALTO server (e.g. the one discovered using [RFC7286]). If this server cannot answer the query itself, it will redirect the client to the "right" ALTO server that has the desired information, using a small to-be-defined extension of the ALTO protocol.

2.3 ALTO clients need to use some kind of "search engine" that indexes ALTO servers and redirects and/or gives cached results.

2.4 ALTO clients need to use a new discovery mechanism to discover the ALTO server that has the desired information and contact it directly.

1.1.2. Discussion of Solution Approaches

The provisioning or initialization protocol for ALTO servers (cf. RFC 5693, Fig 1 [RFC5693]) is currently not standardized. It was a conscious decision not to include this in the scope of the IETF ALTO working group. The reason is that there are many different kinds of information sources. This implementation specific protocol will adapt them to the ALTO server, which offers a standardized protocol to the ALTO clients. However, adding the task of synchronization between ALTO servers to this protocol (i.e., approach 1.1) would overload this protocol with a second functionality that requires standardization for seamless multi-domain operation.

For the 1.? solution approaches, in addition to general technical feasibility and issues like overhead and caching efficiency, another aspect to consider is legal liability. Operator "A" might prefer not to publish information about nodes in or paths between the networks of operators "B" and "C" through A's ALTO server, even if A knew that information. This is not only a question of map size and processing load on A's ALTO server. Operator A could also face legal liability issues if that information had a bad impact on the traffic engineering between B's and C's networks, or on their business models.

No specific actions to build a "search engine" based solution (approach 2.3) are currently known and it is unclear what could be the incentives to operate such an engine. Therefore, this approach is not considered in the remainder of this document.

1.2. The Need for Cross-Domain ALTO Server Discovery

Approaches 1.1, 1.2, 2.1, and 2.2 do not only require the specification of an ALTO protocol extension or a new protocol that runs between ALTO servers. A large-scale, maybe Internet-wide, multi-domain deployment would also need mechanisms by which an ALTO

server could discover other ALTO servers, learn which information is available where, and ideally also who is authorized to publish information related to a given part of the network. Approach 2.4 needs the same mechanisms, except that they are used on the client-side instead of the server-side.

It is sometimes questioned whether there is a need for a solution that allows clients to ask arbitrary queries, even if the ALTO information is partitioned and stored on many ALTO servers. The main argument is, that clients are supposed to optimize the traffic from and to themselves, and that the information needed for that is most likely stored on a "nearby" ALTO server, i.e., the one that can be discovered using [RFC7286]. However, there are scenarios where the ALTO client is not co-located with an endpoint of the to-be-optimized data transmission. Instead, the ALTO client is located at a third party, which takes part in the application signaling, e.g., a so-called "tracker" in a peer-to-peer application. One such scenario, where it is advantageous to place the ALTO client not at an endpoint of the user data transmission, is analyzed in Appendix B.

1.3. Solution Approach

Several solution approaches for cross-domain ALTO server discovery have been evaluated, using the criteria documented in Appendix A. One of them was to use the ALTO protocol itself for the exchange of information availability [I-D.kiesel-alto-alto4alto]. However, the drawback of that approach is that a new registration administration authority would have to be established.

This document specifies a DNS-based procedure for cross-domain ALTO server discovery, which was inspired by "Location Information Server (LIS) Discovery Using IP Addresses and Reverse DNS" [RFC7216]. The primary goal is that this procedure can be used on the client-side (i.e., approach 2.4), but together with new protocols or protocol extensions it could also be used to implement the other solution approaches itemized above.

1.4. ALTO Requirements

During the design phase of the overall ALTO solution, two different server discovery scenarios have been identified and documented in the ALTO requirements document [RFC6708]. The first scenario, documented in Req. AR-32, can be supported using the discovery mechanisms specified in [RFC7286]. An alternative approach, based on IP anycast [I-D.kiesel-alto-ip-based-srv-disc], has also been studied. This document, in contrast, tries to address Req. AR-33.

1.5. Document History

This document is a direct successor of [I-D.kiesel-alto-3pdisc] and [I-D.kist-alto-3pdisc]. The scenario and mechanisms described here and in these documents have been referred to as "third-party server discovery" in the past. However, to avoid naming ambiguities with a completely different scenario, it has been renamed to "ALTO Cross-Domain Server Discovery".

1.6. Feedback

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

2. ALTO Cross-Domain Server Discovery Procedure Specification

2.1. Interface

The algorithm specified in this document takes one IP address "X" and a U-NAPTR [RFC4848] Service Parameter (i.e., "ALTO:http" or "ALTO:https") as parameters. It performs DNS lookups and returns one or more URI(s) of information resources related to that IP address.

For the remainder of the document, we use the notation:
IRD_URIS_X := XDOMDISC(X,"ALTO:https")

2.2. Basic Principle

This algorithm closely follows [RFC7216] and re-uses parts of [RFC7286].

The algorithm sequentially tries two different lookup strategies. First, an ALTO-specific U-NAPTR record is searched in the "reverse tree", i.e., in subdomains of in-addr.arpa. or ip6.arpa. corresponding to the given IP address. If this lookup does not yield a usable result, further lookups with truncated domain names may be tried. The goal is to allow deployment scenarios that require fine-grained discovery on a per-IP basis, as well as large-scale scenarios where discovery is to be enabled for a large number of IP addresses with a small number of additional DNS resource records.

2.3. Step 1: Prepare Domain Name for Reverse DNS Lookup

This task takes the IP address parameter the procedure was called with and constructs a domain name, which is used for DNS lookups in subsequent tasks.

If the IP address given as a parameter to the procedure is an IPv4 address, the domain name is constructed according to the rules specified in Section 3.5 of [RFC1035] and it is rooted in the special domain "IN-ADDR.ARPA.". For IPv6 addresses, the construction rules in Section 2.5 of [RFC3596] apply and the special domain "IP6.ARPA." is used.

Example values for IPv4 and IPv6 addresses could be (Note: a line break was added in the IPv6 example):

R:="3.100.51.198.in-addr.arpa."

R:="0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.B.D.0.1.0.0.2.ip6.arpa."

2.4. Step 2: Add Shortened Domain Names

This task creates a list of several additional domain names, based on the domain name yielded in Step 1.

- For IP version 4, the domain name from Step 1 SHOULD be shortened successively by one and two labels (i.e., purge the first or second dot from the left and everything left of it, respectively), and the results being added to the list. This corresponds to a search on a /24 or /16 network prefix.

- For IP version 6, the domain name from Step 1 SHOULD be shortened successively by 16, 18, 20, and 24 labels, and the results being added to the list. This corresponds to a search on a /64, /56, /48, or /32 network prefix.

This list is intended to provide network operators with a degree of flexibility in where discovery-related resource records can be placed without significantly increasing the number of DNS names that are searched. This does not attach any other significance to these specific zone cuts or create a classful addressing hierarchy based on the reverse DNS tree.

For example, the IPv4 address "192.0.2.75" could result in a list of domain names (with the result from Step 1 put in the first position):

- 75.2.0.192.in-addr.arpa.

- 2.0.192.in-addr.arpa.

- 0.192.in-addr.arpa.

Similarly, the IPv6 address "2001:DB8::28e4:3a93:4429:dfb5" could result in a list:

- 5.b.f.d.9.2.4.4.3.9.a.3.4.e.8.2.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

- 0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

- 0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

- 0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

- 8.b.d.0.1.0.0.2.ip6.arpa.

The limited number of labels by which each name is shortened is intended to limit the maximum number of DNS queries produced by a

single invocation of the cross-domain ALTO server discovery procedure. No more than five U-NAPTR resolutions are invoked for each IP address.

2.5. Step 3: DNS lookups

The list of domain names which was created in the previous step is sequentially (from longest to shortest name) processed, as described in Section 3.2 of RFC 7286 [RFC7286].

3. Using the ALTO Protocol with ALTO Cross-Domain Server Discovery

Based on a modular design principle, ALTO provides several ALTO services, each consisting of a set of information resouces that can be accessed using the ALTO protocol. The ALTO protocol specification defines the following ALTO services and their corresponding information resouces:

- Network and Cost Map Service, see Section 11.2 of [RFC7285]
- Map-Filtering Service, see Section 11.3 of [RFC7285]
- Endpoint Property Service, see Section 11.4 of [RFC7285]
- Endpoint Cost Service, see Section 11.5 of [RFC7285]

Extension documents may specify further information resources; however, these are out of scope of this document. The information resources that are available at a specific ALTO server are listed in its Information Resource Directory (IRD, see Section 9 of [RFC7285]).

3.1. Endpoint Property Service

If an ALTO client wants to query an Endpoint Property Service (see Section 11.4 of RFC 7285 [RFC7285]) about an endpoint with IP address "X", it has to perform the following steps:

1. Invoke the ALTO Cross-Domain Server Discovery Procedure (as specified in Section 2): IRD_URIS_X := XDOMDISC(X,"ALTO:https")
2. The result IRD_URIS_X is a list of one or more Information Resource Directories (IRD, see Section 9 of [RFC7285]). Check each of these IRDs for a suitable Endpoint Property Service and query it.

If the ALTO client wants to do a similar Endpoint Property query for a different IP address "Y", the whole procedure has to be repeated, as IRD_URIS_Y := XDOMDISC(Y,"ALTO:https") may yield a different list of IRDs. Of course, the results of individual DNS queries may be cached as indicated by their respective time-to-live (TTL) values.

3.2. Endpoint Cost Service

The ALTO Endpoint Cost Service (ECS, see Section 11.5 of RFC 7285 [RFC7285]) provides information about costs between individual endpoints and it also supports ranking. The ECS allows that endpoints may be denoted by IP prefixes or IP addresses (as special case of a prefix); however, this document assumes that all endpoints

are single IP addresses. The ECS is called with a list of one or more source IP addresses, which we will call (S1, S2, S3, ...), and a list of one or more destination IP addresses, which we will call (D1, D2, D3, ...).

This specification distinguishes several cases, regarding the number of elements in the list of source and destination addresses, respectively:

1. Exactly one source address S1 and more than one destination addresses D1, D2, D3, ... In this case, the ALTO client has to perform the following steps:

    1. Invoke the ALTO Cross-Domain Server Discovery Procedure (as specified in Section 2):
    IRD_URIS_S1 := XDOMDISC(S1,"ALTO:https")

    2. The result IRD_URIS_S1 is a list of one or more Information Resource Directories (IRD, see Section 9 of [RFC7285]). Check each of these IRDs for a suitable ECS and query it.

2. More than one source addresses S1, S2, S3, ... and exactly one destination address D1. In this case, the ALTO client has to perform the following steps:

    1. Invoke the ALTO Cross-Domain Server Discovery Procedure (as specified in Section 2):
    IRD_URIS_D1 := XDOMDISC(D1,"ALTO:https")

    2. The result IRD_URIS_D1 is a list of one or more Information Resource Directories (IRD, see Section 9 of [RFC7285]). Check each of these IRDs for a suitable ECS and query it.

3. Exactly one source address S1 and exactly one destination address D1. The ALTO client may perform the same steps as in case 1, as specified above. As an alternative, it may also perform the same steps as in case 2, as specified above.

4. More than one source addresses S1, S2, S3, ... and more than one destination addresses D1, D2, D3, ... In this case, the ALTO client should split the list of source addresses, and perform separately for each source address the same steps as in case 1, as specified above. As an alternative, the ALTO client could also split the list of destination addresses, and perform separately for each destination address the same steps as in case 2, as specified above.

3.3. Network and Cost Map Service

An ALTO client may invoke the ALTO Cross-Domain Server Discovery Procedure (as specified in Section 2) for an IP address "X" and get a list of one or more IRD URI(s): IRD_URIS_X := XDOMDISC(X,"ALTO:https"). These IRD(s) will always contain a network and a cost map, as these are mandatory information ressources (see Section 11.2 of [RFC7285]). However, the cost matrix may be very sparse. If, according to the network map, PID_X is the PID that contains the IP address X, and PID_1, PID_2, PID_3, ... are other PIDS, the cost map may look like this:

From \ To	PID_1	PID_2	PID_X	PID_3
PID_1			92	
PID_2			6	
PID_X	46	3	1	19
PID_3			38	

In this example, all cells outside column "X" and row "X" are unspecified. A cost map with this structure contains the same information as what could be retrieved using the ECS, cases 1 and 2 in the previous subsection. Accessing cells outside column "X" and row "X" may not yield useful results.

Trying to assemble a more densely populated cost map from several cost maps with this very sparse structure may be a non-trivial task, as different ALTO servers may use different PID definitions (i.e., network maps) and incompatible scales for the costs, in particular for the "routingcost" metric.

3.4. Map-Filtering Service

This issue is left for further study in this version of the draft.

4. Implementation, Deployment, and Operational Considerations

4.1. Considerations for ALTO Clients

4.1.1. Resource Consumer Initiated Discovery

To some extent, ALTO requirement AR-32 [RFC6708], i.e., resource consumer initiated ALTO server discovery, can be seen as a special case of cross-domain ALTO server discovery. To that end, an ALTO client embedded in a resouce consumer would have to figure out its own "public" IP address and perform the procedures described in this document on that address. However, due to the widespread deployment of Network Address Translators (NAT), additional protocols and mechanisms such as STUN [RFC5389] would be needed and considerations for UNSAF [RFC3424] apply. Therefore, using the procedures specified in this document for resource consumer based ALTO server discovery is generally NOT RECOMMENDED. Note that a less versatile yet simpler approach for resource consumer initiated ALTO server discovery is specified in [RFC7286].

4.1.2. IPv4/v6 Dual Stack, Multihoming, NAT, and Host Mobility

The algortihm specified in this document can discover ALTO server URIs for a given IP address. The intention is, that a third party (e.g., a resource directory) that receives query messages from a resource consumer can use the source address in these messages to discover suitable ALTO servers for this specific resource consumer.

However, resource consumers (as defined in Section 2 of [RFC5693]) may reside on hosts with more than one IP address, e.g., due to IPv4/v6 dual stack operation and/or multihoming. IP packets sent with different source addresses may be subject to different routing policies and path costs. In some deployment scenarios, it may even be required to ask different sets of ALTO servers for guidance. Furthermore, source addresses in IP packets may be modified en-route by Network Address Translators (NAT).

If a resource consumer queries a resource directory for candidate resource providers, the locally selected (and possibly en-route translated) source address of the query message - as observed by the resource directory - will become the basis for the ALTO server discovery and the subsequent optimization of the resource directory's reply. If, however, the resource consumer then selects different source addresses to contact returned resource providers, the desired better-than-random "ALTO effect" may not occur.

Therefore, a dual stack or multihomed resource consumer SHOULD either always use the same address for contacting the resource directory and

the resource providers, i.e., overriding the operating system's automatic source IP address selection, or use resource consumer based ALTO server discovery [RFC7286] to discover suitable ALTO servers for every local address and then locally perform ALTO-influenced resource consumer selection and source address selection. Similarly, resource consumers on mobile hosts SHOULD query the resource directory again after a change of IP address, in order to get a list of candidate resource providers that is optimized for the new IP address.

4.2. Deployment Considerations for Network Operators

4.2.1. Separation of Interests

We assume that if two organizations share parts of their DNS infrastructure, i.e., have common in-addr.arpa. and/or ip6.arpa. subdomains, they will also be able to operate a common ALTO server, which still may do redirections if desired or required by policies.

Note that the ALTO server discovery procedure is supposed to produce only a first URI of an ALTO server that can give reasonable guidance to the client. An ALTO server can still return different results based on the client's address (or other identifying properties) or redirect the client to another ALTO server using mechanisms of the ALTO protocol (see Sect. 9 of [RFC7285]).

5. Security Considerations

A high-level discussion of security issues related to ALTO is part of the ALTO problem statement [RFC5693]. A classification of unwanted information disclosure risks, as well as specific security-related requirements can be found in the ALTO requirements document [RFC6708].

The remainder of this section focuses on security threats and protection mechanisms for the cross-domain ALTO server discovery procedure as such. Once the ALTO server's URI has been discovered and the communication between the ALTO client and the ALTO server starts, the security threats and protection mechanisms discussed in the ALTO protocol specification [RFC7285] apply.

5.1. Integrity of the ALTO Server's URI

Scenario Description

An attacker could compromise the ALTO server discovery procedure or infrastructure in a way that ALTO clients would discover a "wrong" ALTO server URI.

Threat Discussion

This is probably the most serious security concern related to ALTO server discovery. The discovered "wrong" ALTO server might not be able to give guidance to a given ALTO client at all, or it might give suboptimal or forged information. In the latter case, an attacker could try to use ALTO to affect the traffic distribution in the network or the performance of applications (see also Section 15.1. of [RFC7285]). Furthermore, a hostile ALTO server could threaten user privacy (see also Section 5.2.1, case (5a) in [RFC6708]).

However, it should also be noted that, if an attacker was able to compromise the DNS infrastructure used for cross-domain ALTO server discovery, (s)he could also launch significantly more serious other attacks (e.g., redirecting various application protocols).

Protection Strategies and Mechanisms

The cross-domain ALTO server discovery procedure relies on a series of DNS lookups. If an attacker was able to modify or spoof any of the DNS records, the resulting URI could be replaced by a forged URI. The application of DNS security (DNSSEC) [RFC4033] provides a means to limit attacks that rely on modification of the DNS records while in transit. Additional operational precautions for safely operating the DNS infrastructure are required in order to ensure that name servers do not sign forged (or otherwise

"wrong") resource records. Security considerations specific to U-NAPTR are described in more detail in [RFC4848].

A related risk is the impersonation of the ALTO server (i.e., attacks after the correct URI has been discovered). This threat and protection strategies are discussed in Section 15.1 of [RFC7285]. Note that if TLS is used to protect ALTO, the server certificate will contain the host name (CN). Consequently, only the host part of the HTTPS URI will be authenticated, i.e., the result of the ALTO server discovery procedure. The DNS/U-NAPTR based mapping within the cross-domain ALTO server discovery procedure needs to be secured as described above, e.g., by using DNSSEC.

In addition to active protection mechanisms, users and network operators can monitor application performance and network traffic patterns for poor performance or abnormalities. If it turns out that relying on the guidance of a specific ALTO server does not result in better-than-random results, the usage of the ALTO server may be discontinued (see also Section 15.2 of [RFC7285]).

5.2. Availability of the ALTO Server Discovery Procedure

Scenario Description

An attacker could compromise the cross-domain ALTO server discovery procedure or infrastructure in a way that ALTO clients would not be able to discover any ALTO server.

Threat Discussion

If no ALTO server can be discovered (although a suitable one exists) applications have to make their decisions without ALTO guidance. As ALTO could be temporarily unavailable for many reasons, applications must be prepared to do so. However, The resulting application performance and traffic distribution will correspond to a deployment scenario without ALTO.

Protection Strategies and Mechanisms

Operators should follow best current practices to secure their DNS and ALTO (see Section 15.5 of [RFC7285]) servers against Denial-of-Service (DoS) attacks.

5.3. Confidentiality of the ALTO Server's URI

Scenario Description

An unauthorized party could invoke the cross-domain ALTO server discovery procedure, or intercept discovery messages between an authorized ALTO client and the DNS servers, in order to acquire knowledge of the ALTO server URI for a specific IP address.

Threat Discussion

In the ALTO use cases that have been described in the ALTO problem statement [RFC5693] and/or discussed in the ALTO working group, the ALTO server's URI as such has always been considered as public information that does not need protection of confidentiality.

Protection Strategies and Mechanisms

No protection mechanisms for this scenario have been provided, as it has not been identified as a relevant threat. However, if a new use case is identified that requires this kind of protection, the suitability of this ALTO server discovery procedure as well as possible security extensions have to be re-evaluated thoroughly.

5.4. Privacy for ALTO Clients

Scenario Description

An unauthorized party could intercept messages between an ALTO client and the DNS servers, and thereby find out the fact that said ALTO client uses (or at least tries to use) the ALTO service in order to optimize traffic from/to a specific IP address.

Threat Discussion

In the ALTO use cases that have been described in the ALTO problem statement [RFC5693] and/or discussed in the ALTO working group, this scenario has not been identified as a relevant threat.

Protection Strategies and Mechanisms

No protection mechanisms for this scenario have been provided, as it has not been identified as a relevant threat. However, if a new use case is identified that requires this kind of protection, the suitability of this ALTO server discovery procedure as well as possible security extensions have to be re-evaluated thoroughly.

6. IANA Considerations

This document does not require any IANA action.

7. References

7.1. Normative References

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

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

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[RFC4848] Daigle, L., "Domain-Based Application Service Location Using URIs and the Dynamic Delegation Discovery Service (DDDS)", RFC 4848, April 2007.

7.2. Informative References

[I-D.ietf-alto-deployments]
Stiemerling, M., Kiesel, S., Scharf, M., Seidel, H., and S. Previdi, "ALTO Deployment Considerations", draft-ietf-alto-deployments-15 (work in progress), May 2016.

[I-D.kiesel-alto-3pdisc]
Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., Tomsu, M., and H. Song, "ALTO Server Discovery Protocol", draft-kiesel-alto-3pdisc-05 (work in progress), March 2011.

[I-D.kiesel-alto-alto4alto]
Kiesel, S., "Using ALTO for ALTO server selection", draft-kiesel-alto-alto4alto-00 (work in progress), July 2010.

[I-D.kiesel-alto-ip-based-srv-disc]
Kiesel, S. and R. Penno, "Application-Layer Traffic Optimization (ALTO) Anycast Address", draft-kiesel-alto-ip-based-srv-disc-03 (work in progress), July 2014.

[I-D.kist-alto-3pdisc]
Kiesel, S., Krause, K., and M. Stiemerling, "Third-Party ALTO Server Discovery (3pdisc)", draft-kist-alto-3pdisc-05 (work in progress), January 2014.

[RFC3424] Daigle, L. and IAB, "IAB Considerations for UNilateral Self-Address Fixing (UNSAF) Across Network Address Translation", RFC 3424, November 2002.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.

[RFC6708] Kiesel, S., Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", RFC 6708, September 2012.

[RFC7216] Thomson, M. and R. Bellis, "Location Information Server (LIS) Discovery Using IP Addresses and Reverse DNS", RFC 7216, April 2014.

[RFC7285] Alimi, R., Penno, R., Yang, Y., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, September 2014.

[RFC7286] Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., and H. Song, "Application-Layer Traffic Optimization (ALTO) Server Discovery", RFC 7286, June 2014.

Appendix A. Requirements for ALTO Cross-Domain Server Discovery

A solution for the problem described in the previous section would be an ALTO Cross-Domain Server Discovery system. This section itemizes requirements.

A.1. Discovery Client Application Programming Interface

The discovery client will be called through some kind of application programming interface (API) and the parameters will be an IP address and, for purposes of extensibility, a service identifier such as "ALTO". It will return one or more URI(s) that offers the requested service ("ALTO") for the given IP address.

In other words, the client would be used to retrieve a mapping:

(IP address, "ALTO") -> IRD-URI(s)

where IRD-URI(s) is one or more URI(s) of Information Resource Directories (IRD, see Section 9 of [RFC7285]) of ALTO server(s) that can give reasonable guidance to a resource consumer with the indicated IP address.

A.2. Data Storage and Authority Requirements

The information for mapping IP addresses and service parameters to URIs should be stored in a - preferably distributed - database. It must be possible to delegate administration of parts of this database. Usually, the mapping from a specific IP address to an URI is defined by the authority that has administrative control over this IP address, e.g., the ISP in residential access networks or the IT department in enterprise, university, or similar networks.

A.3. Cross-Domain Operations Requirements

The cross-domain server discovery mechanism should be designed in such a way that it works across the public Internet and also in other IP-based networks. This in turn means that such mechanisms cannot rely on protocols that are not widely deployed across the Internet or protocols that require special handling within participating networks. An example is multicast, which is not generally available across the Internet.

The ALTO Cross-Domain Server Discovery protocol must support gradual deployment without a network-wide flag day. If the mechanism needs some kind of well-known "rendezvous point", re-using an existing infrastructure (such as the DNS root servers or the WHOIS database) should be preferred over establishing a new one.

A.4. Protocol Requirements

The protocol must be able to operate across middleboxes, especially across NATs and firewalls.

The protocol shall not require any pre-knowledge from the client other than any information that is known to a regular IP host on the Internet.

A.5. Further Requirements

The ALTO cross domain server discovery cannot assume that the server discovery client and the server discovery responding entity are under the same administrative control.

Appendix B. ALTO and Tracker-based Peer-to-Peer Applications

The ALTO protocol specification [RFC7285] details how an ALTO client can query an ALTO server for guiding information and receive the corresponding replies. However, in the considered scenario of a tracker-based P2P application, there are two fundamentally different possibilities where to place the ALTO client:

1. ALTO client in the resource consumer ("peer")

2. ALTO client in the resource directory ("tracker")

In the following, both scenarios are compared in order to explain the need for ALTO queries on behalf of remote resource consumers.

In the first scenario (see Figure 2), the resource consumer queries the resource directory for the desired resource (F1). The resource directory returns a list of potential resource providers without considering ALTO (F2). It is then the duty of the resource consumer to invoke ALTO (F3/F4), in order to solicit guidance regarding this list.

In the second scenario (see Figure 4), the resource directory has an embedded ALTO client. After receiving a query for a given resource (F1) the resource directory invokes this ALTO client to evaluate all resource providers it knows (F2/F3). Then it returns a, possibly shortened, list containing the "best" resource providers to the resource consumer (F4).

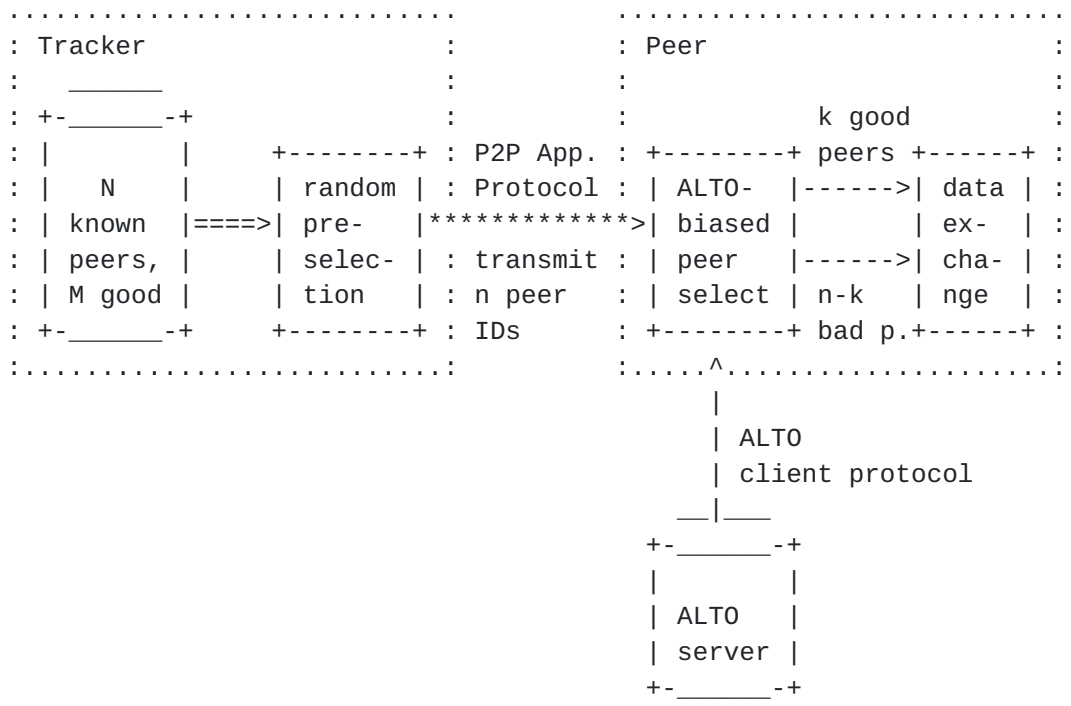


Figure 1: Tracker-based P2P Application with random peer preselection

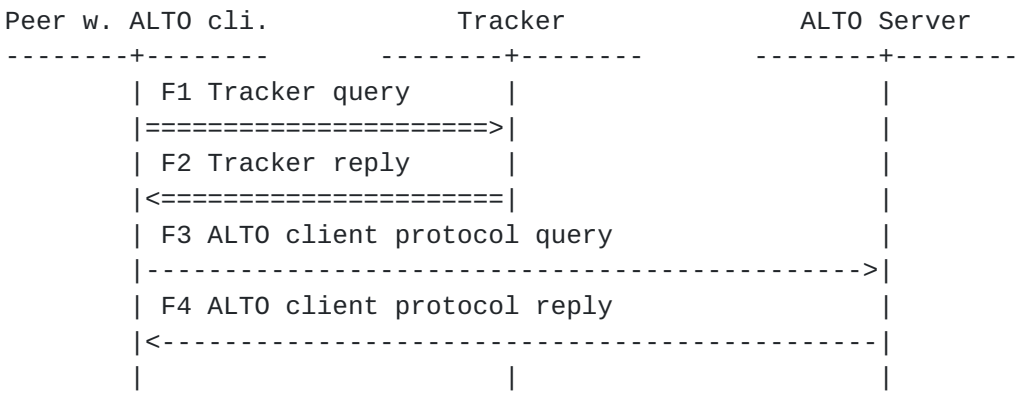


Figure 2: Basic message sequence chart for resource consumer-initiated ALTO query

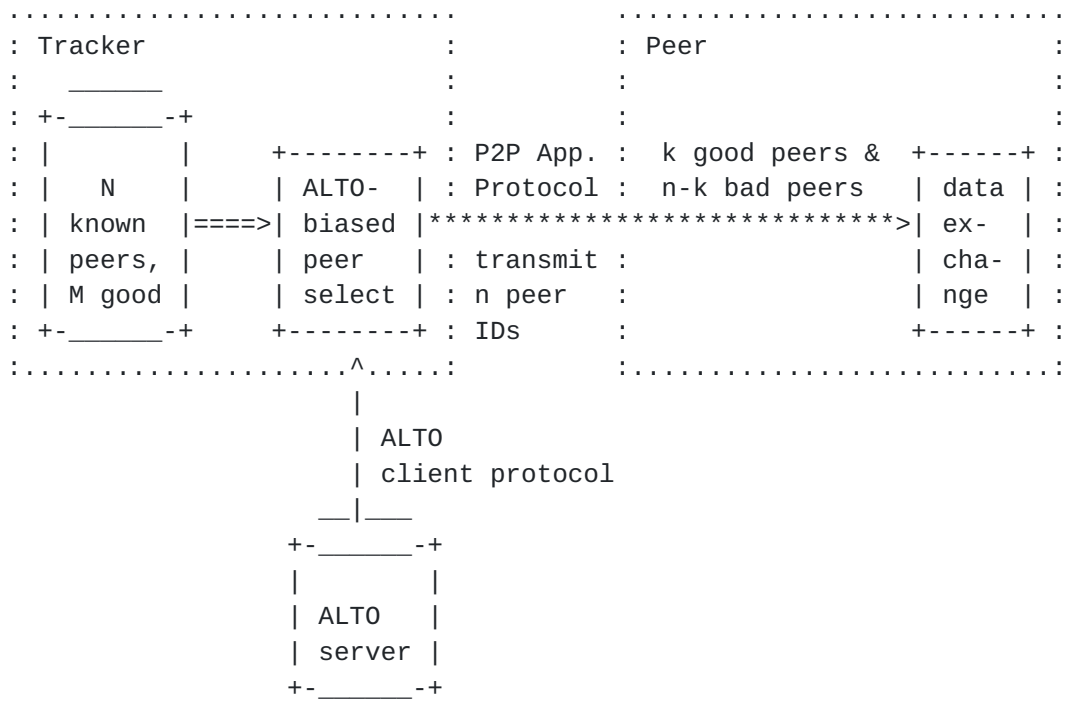
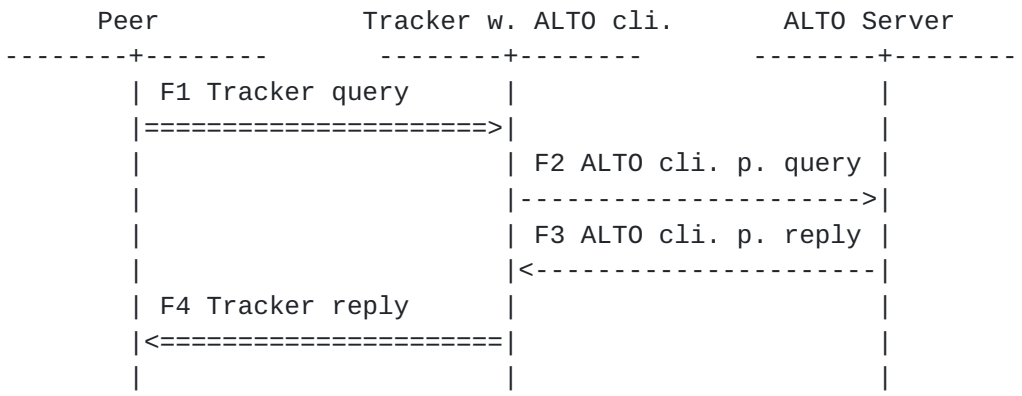


Figure 3: Tracker-based P2P Application with ALTO client in tracker



==== Application protocol (i.e., tracker-based P2P app protocol)
---- ALTO client protocol

Figure 4: Basic message sequence chart for ALTO query on behalf of remote resource consumer

Note: the message sequences depicted in Figure 2 and Figure 4 may occur both in the target-aware and the target-independent query mode (c.f. [RFC6708]). In the target-independent query mode no message exchange with the ALTO server might be needed after the tracker query, because the candidate resource providers could be evaluated using a locally cached "map", which has been retrieved from the ALTO

server some time ago.

The problem with the first approach is, that while the resource directory might know thousands of peers taking part in a swarm, the list returned to the resource consumer is usually shortened for efficiency reasons. Therefore, the "best" (in the sense of ALTO) potential resource providers might not be contained in that list anymore, even before ALTO can consider them.

For illustration, consider a simple model of a swarm, in which all peers fall into one of only two categories: assume that there are "good" ("good" in the sense of ALTO's better-than-random peer selection, based on an arbitrary desired rating criterion) and "bad' peers only. Having more different categories makes the maths more complex but does not change anything to the basic outcome of this analysis. Assume that the swarm has a total number of N peers, out of which are M "good" and N-M "bad" peers, which are all known to the tracker. A new peer wants to join the swarm and therefore asks the tracker for a list of peers.

If, according to the first approach, the tracker randomly picks n peers from the N known peers, the result can be described with the hypergeometric distribution. The probability that the tracker reply contains exactly k "good" peers (and n-k "bad" peers) is:

$$P(X=k) = \frac{\binom{m}{k}\binom{N-m}{n-k}}{\binom{N}{n}}$$

$$\text{with} \quad \binom{n}{k} = \frac{n!}{k!\,(n-k)!} \quad \text{and} \quad n! = n * (n-1) * (n-2) * .. * 1$$

The probability that the reply contains at most k "good" peers is: $P(X<=k)=P(X=0)+P(X=1)+..+P(X=k)$.

For example, consider a swarm with N=10,000 peers known to the tracker, out of which M=100 are "good" peers. If the tracker randomly selects n=100 peers, the formula yields for the reply: $P(X=0)=36\%$, $P(X<=4)=99\%$. That is, with a probability of approx. 36% this list does not contain a single "good" peer, and with 99% probability there are only four or less of the "good" peers on the

list. Processing this list with the guiding ALTO information will ensure that the few favorable peers are ranked to the top of the list; however, the benefit is rather limited as the number of favorable peers in the list is just too small.

Much better traffic optimization could be achieved if the tracker would evaluate all known peers using ALTO, and return a list of 100 peers afterwards. This list would then include a significantly higher fraction of "good" peers. (Note, that if the tracker returned "good" peers only, there might be a risk that the swarm might disconnect and split into several disjunct partitions. However, finding the right mix of ALTO-biased and random peer selection is out of the scope of this document.)

Therefore, from an overall optimization perspective, the second scenario with the ALTO client embedded in the resource directory is advantageous, because it is ensured that the addresses of the "best" resource providers are actually delivered to the resource consumer. An architectural implication of this insight is that the ALTO server discovery procedures must support ALTO queries on behalf of remote resource consumers. That is, as the tracker issues ALTO queries on behalf of the peer which contacted the tracker, the tracker must be able to discover an ALTO server that can give guidance suitable for that respective peer.

Appendix C. Contributors List and Acknowledgments

The initial version of this document was co-authored by Marco Tomsu (Alcatel-Lucent).

This document borrows some text from [RFC7286], as it was historically part of that memo. Special thanks to Michael Scharf and Nico Schwan.

Authors' Addresses

Sebastian Kiesel
University of Stuttgart Information Center
Allmandring 30
Stuttgart 70550
Germany

Email: ietf-alto@skiesel.de
URI: http://www.rus.uni-stuttgart.de/nks/

Martin Stiemerling
University of Applied Sciences Darmstadt, Computer Science Dept.
Haardtring 100
Darmstadt 64295
Germany

Phone: +49 6151 16 7938
Email: mls.ietf@gmail.com
URI: http://ietf.stiemerling.org