ALTO
Internet-Draft
Intended status: Experimental
Expires: July 17, 2014
S. Kiesel
K. Krause
University of Stuttgart
M. Stiemerling
NEC Europe Ltd.
January 13, 2014

Third-Party ALTO Server Discovery (3pdisc)
draft-kist-alto-3pdisc-05

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.

This document specifies a procedure for third-party ALTO server discovery, which can be used if the ALTO client is not co-located with the actual resource consumer, but instead embedded in a third party such as a peer-to-peer tracker.

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 several DNS lookups (for U-NAPTR and SOA resource records) 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

This Internet-Draft will expire on July 17, 2014.

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 a client-server protocol; see requirement AR-1 in [RFC6708]. Before an ALTO client can ask for guidance it needs to discover one or more ALTO servers that can provide suitable guidance. For applications that use a centralized resource directory, such as tracker-based P2P applications, the efficiency of ALTO is significantly improved if the ALTO client is embedded in said resource directory instead of the resource consumer (see Appendix A for a detailed example and analysis of such a scenario). The ALTO client embedded into the resource directory asks for guidance on behalf of the resource consumers. To that end, it needs to discover ALTO servers that can give guidance suitable for these resource consumers, respectively. This is called third-party party ALTO server discovery.

This document specifies a procedure for third-party ALTO server discovery. In other words, this document tries to meet requirement AR-33 in [RFC6708].

The ALTO protocol specification [I-D.ietf-alto-protocol] is based on HTTP and expects the discovery procedure to yield the HTTP(S) URI of an ALTO server's information resource directory. Therefore, this document specifies an algorithm that takes a resource consumer's IP address as argument, performs several DNS lookups (for U-NAPTR [RFC4848] and SOA resource records), and produces URIs of ALTO servers that are able to give reasonable ALTO guidance to a resource consumer willing to communicate using this IP address.

To some extent, AR-32, i.e., resource consumer initiated ALTO server discovery, can be seen as a special case of third-party ALTO server discovery. However, the considerations in Section 3.1.1 apply. Note that a less versatile yet simpler approach for resource consumer initiated ALTO server discovery is specified in [I-D.ietf-alto-server-discovery].

A more detailed discussion of various options where to place the functional entities comprising the overall ALTO architecture can be found in [I-D.ietf-alto-deployments].

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

2. Third-party ALTO Server Discovery Procedure Specification

2.1. Interface

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 several DNS lookups (for U-NAPTR and SOA resource records) and returns one or more URI(s) of information resources related to that IP address.

2.2. Basic Principle

The algorithm sequentially tries two different lookup strategies. First, an ALTO-specific U-NAPTR lookup is performed in the "reverse tree", i.e., in subdomains of in-addr.arpa. or ip6.arpa., respectively. If this lookup does not yield a usable result, the SOA record for the reverse zone is acquired, its master name server (MNAME) value is extracted and used for a further ALTO-specific U-NAPTR lookup.

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. Overall Procedure

This figure gives an overview on the third-party discovery procedure. All tasks (T) and conditional branches (B) are specified below.

```
                (---------------------------------------)
                ( START 3pdisc with parameters          )
                ( IP_address IP, Service_Parameter SP   )
                (-------------------+-------------------)
                                    V
                +- T1 --------------+-------------------+
                | R:=<IP>.in-addr.arpa. / <IP>.ip6.arpa.|
                +-------------------+-------------------+
                                    V
                +- T2 --------------+-------------------+
                | X:=DNSlookup(R,U-NAPTR,SP)            |
                +-------------------+-------------------+
                                    V
                 / B1 --------------+------------------\
      /---------< One or more U-NAPTR results in X      >
      |      yes \------------------+------------------/
      |                             V no
      |          /- B2 -------------+------------------\
      |    /----< Authority sect. with SOA record in X  >
      |    | yes \------------------+------------------/
      |    |                        V no
      |    |    +- T3 --------------+-------------------+
      |    |    | X:=DNSlookup(R,SOA)                   |
      |    |    +-------------------+-------------------+
      |    |                        V
      |    |     /- B3 -------------+------------------\
      |    |    < Lookup OK, SOA record present in X    >----\
      |    |     \------------------+------------------/ no  |
      |    |                        V yes                    |
      |    \----------------------->+                        |
      |                             V                        |
      |         +- T4 --------------+-------------------+    |
      |         | M:=extract MNAME from SOA record in X |    |
      |         | X:=DNSlookup(M,U-NAPTR,SP)            |    |
      |         +-------------------+-------------------+    |
      |                             V                        |
      |          /- B4 -------------+------------------\     V
      \--->+<---< One or more U-NAPTR results in X      >--->+
           | yes \-------------------------------------/ no  |
           V                                                 V
   (-------+-------)                                 (-------+-------)
   ( END, result X )                                 ( END, failure  )
   (---------------)                                 (---------------)
```

2.4. Specification of Tasks and Conditional Branches

2.4.1. T1: Prepare Domain Name for Reverse DNS Lookup

Task T1 takes the IP address parameter the 3pdisc procedure was called with and constructs a domain name, which is stored in variable "R" for use in subsequent tasks.

If the IP address given as a parameter to the 3pdisc 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 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 "R" 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:="0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.B.D.0.
1.0.0.2.ip6.arpa."
```

2.4.2. T2/B1: U-NAPTR Lookup in Reverse Zone

Task T1 performs a U-NAPTR lookup as specified in [RFC4848] on "R", in order to get service-specific U-NAPTR resource records that are directly associated with the IP address in question.

The ALTO protocol specification defines HTTP and HTTPS as transport mechanisms and URI schemes for ALTO. Consequently, the U-NAPTR lookup is performed with the "ALTO" Application Service Tag and either the "http" or the "https" Application Protocol Tag. Application Service Tag and Application Protocol Tag are concatenated to form the Service Parameter SP, i.e., either "ALTO:http" or "ALTO:https".

The goal of said U-NAPTR lookup is to obtain one or more URIs for the ALTO server's Information Resource Directory. If two or more URIs are found they are sorted according to their order and preference fields as specified in [RFC4848] and [RFC3403].

The lookup result, including a SOA record that may or may not be present in the authority section, is stored in variable "X".

As an example, the following two U-NAPTR resource records can be used for mapping "3.100.51.198.in-addr.arpa." to the HTTPS URI

https://altoserver.isp.example.net/secure/directory or the HTTP URI http://altoserver.isp.example.net/directory, with the former being preferred.

```
3.100.51.198.in-addr.arpa.

IN NAPTR 100  10   "u"    "ALTO:https"
     "!.*!https://altoserver.isp.example.net/secure/directory!"  ""

IN NAPTR 200  10   "u"    "ALTO:http"
     "!.*!http://altoserver.isp.example.net/directory!"  ""
```

Conditional Branch B1 checks whether at least one U-NAPTR record matching the service parameter SP could be retrieved. If so, the procedure ends successfully and the sorted list of U-NAPTR records is the result. Otherwise, if no U-NAPTR records could be retrieved, we continue with B2.

Note: The U-NAPTR lookup in Task T2 is identical to Step 2 specified in [I-D.ietf-alto-server-discovery], which specifies with "manual input" and "DHCP" two alternatives for acquiring the name to be looked up. Therefore, it is possible to merge both documents into a common ALTO server discovery framework.

2.4.3. B2/T3/B3: Acquire SOA Record for Reverse Zone

The task of B2/T3/B3 is to acquire the SOA record for the "reverse zone", i.e., the zone in the in-addr.arpa. or ip6.arpa. domain that contains the IP address in question.

A sample SOA record could be:

```
100.51.198.in-addr.arpa
IN  SOA dns1.isp.example.net.   hostmaster.isp.example.net. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
```

Conditional Branch B2 checks whether the SOA record was present in the authority section of X, i.e., the result of Task T2. If not, an explicit lookup is done in Task T3. If Conditional Branch B3 determines that this explicit lookup failed, the discovery procedure is aborted without a result; otherwise we continue with T4.

2.4.4. T4/B4: U-NAPTR Lookup on SOA-MNAME

Now that the SOA record is available, Task T4 first extracts the MNAME field, i.e., the responsible master name server from the SOA record. An example MNAME could be:

    dns1.isp.example.net.

Then, a U-NAPTR lookup as specified in Task T2 is performed on this MNAME and the result is stored in variable "X".

Conditional Branch B4 checks whether at least one U-NAPTR record matching the service parameter SP could be retrieved. If so, the procedure ends successfully and the sorted list of U-NAPTR records is the result. Otherwise, if no U-NAPTR records could be retrieved, the discovery procedure is aborted without a result.

3. Implementation, Deployment, and Operational Considerations

3.1. Considerations for ALTO Clients

3.1.1. 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 third-party 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 [I-D.ietf-alto-server-discovery].

3.1.2. 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 [I-D.ietf-alto-server-discovery] 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.

3.2. Deployment Considerations for Network Operators

3.2.1. NAPTR in Reverse Tree vs. SOA-based discovery

As already outlined in Section 2.2, the third-party discovery procedure sequentially tries two different lookup strategies, thus giving network operators the choice of two different deployment options:

- o Individual NAPTR records in the in-addr.arpa or ip6.arpa domains allow very fine-grained discovery of ALTO "entry point" URIs on a per-IP-address basis. This method also gives the fastest response times and causes a comparatively low load on the DNS, as the algorithm terminates successfully after the first DNS query. DNS operators that already maintain reverse zones (e.g., for PTR records) should prefer this option, possibly using DNS server implementation-specific methods for mass deployment (e.g., BIND9's $GENERATE statement).

- o If a DNS operator considers the first option too cumbersome, or if IPv6 privacy extensions is to be used without dynamic PTR updates, setting up SOA records in the in-addr.arpa. or ip6.arpa. subdomains plus setting up corresponding ALTO-specific U-NAPTR records will also give reasonable, yet less fine-grained results at the cost of slightly higher delay and load on the DNS.

3.2.2. Separation of Interests

We assume that if two organizations share parts of their DNS infrastructure, i.e., have a common SOA record in their in-addr.arpa. or ip6.arpa. subdomain(s), 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. 6.7 of [I-D.ietf-alto-protocol]).

3.3. Impact on DNS

3.3.1. Non-PTR Resource Records in Reverse Tree

Installing NAPTR records, i.e., a record type other than PTR records, in the in-addr.arpa or ip6.arpa domain may seem uncommon, but it is not a new concept. Earlier documents that specify the usage of Non-PTR resource records in the reverse tree include RFC 4025 [RFC4025], RFC 4255 [RFC4255], and RFC 4322 [RFC4322].

3.3.2. Usage with DNS Hidden Master Servers

In some deployment scenarios, the Master DNS server for a in-addr.arpa. or ip6.arpa. subdomain, as indicated in the respective SOA record, may not be reachable due to traffic restrictions ("hidden master"). This does not cause any problems with the algorithm described here, as the MNAME is only used for further DNS lookups; but it is never attempted to contact this server directly.

3.3.3. Load on the DNS

The procedure described in this document features several nested conditional branches, but no loops. Each time being called it attempts one to three DNS lookups.

4. 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 third-party 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 [I-D.ietf-alto-protocol] apply.

4.1. 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 14.1. of [I-D.ietf-alto-protocol]). 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 third-party ALTO server discovery (see below), (s)he could also launch significantly more serious other attacks (e.g., redirecting various application protocols).

Protection Strategies and Mechanisms
The third-party 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 14.1 of [I-D.ietf-alto-protocol]. 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 third-party 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 14.2 of [I-D.ietf-alto-protocol]).

4.2. Availability of the ALTO Server Discovery Procedure

Scenario Description
An attacker could compromise the third-party 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 14.5 of [I-D.ietf-alto-protocol]) servers against Denial-of-Service (DoS) attacks.

4.3. Confidentiality of the ALTO Server's URI

Scenario Description

An unauthorized party could invoke the third-party 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 resource consumer.

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.

4.4. 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 on behalf of a specific resource consumer.

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.

5. IANA Considerations

This document does not require any IANA action.

This document specifies an algorithm that uses U-NAPTR lookups [RFC4848] with the Application Service Tag "ALTO" and the Application Protocol Tags "http" and "https". These tags have already been registered with IANA. In particular, for the registration of the Application Service Tag "ALTO", see [I-D.ietf-alto-server-discovery].

6. References

6.1. 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.

[RFC3403] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

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

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

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

6.2. Informative References

[I-D.ietf-alto-deployments]
Stiemerling, M., Kiesel, S., Previdi, S., and M. Scharf, "ALTO Deployment Considerations", draft-ietf-alto-deployments-08 (work in progress), October 2013.

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-21 (work in progress), November 2013.

[I-D.ietf-alto-server-discovery]
Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., and H. Song, "ALTO Server Discovery", draft-ietf-alto-server-discovery-10 (work in progress), September 2013.

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

[RFC4025] Richardson, M., "A Method for Storing IPsec Keying Material in DNS", RFC 4025, March 2005.

[RFC4255] Schlyter, J. and W. Griffin, "Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints", RFC 4255, January 2006.

[RFC4322] Richardson, M. and D. Redelmeier, "Opportunistic Encryption using the Internet Key Exchange (IKE)", RFC 4322, December 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.

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

The ALTO protocol specification [I-D.ietf-alto-protocol] 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 third-party ALTO queries.

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, which we will refer to as 3PAC (Third-Party ALTO Client) in this document. After receiving a query for a given resource (F1) the resource directory invokes the 3PAC 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).

```
.............................          .............................
: Tracker                    :          : Peer                        :
:   ______                   :          :                             :
: +-______-+                 :          :            k good           :
: |        |     +--------+ : P2P App. : +--------+ peers +------+ :
: |   N    |     | random | : Protocol : | ALTO-  |------>| data | :
: | known  |====>| pre-   |*************>| biased |       | ex-  | :
: | peers, |     | selec- | : transmit : | peer   |------>| cha- | :
: | M good |     | tion   | : n peer   : | select | n-k    | nge  | :
: +-______-+     +--------+ : IDs      : +--------+ bad p.+------+ :
:...........................:          :.....^.....................:
                                             |
                                             | ALTO
                                             | client protocol
                                           __|___
                                         +-______-+
                                         |        |
                                         | ALTO   |
                                         | server |
                                         +-______-+
```

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

```
Peer w. ALTO cli.            Tracker               ALTO Server
--------+--------       --------+--------       --------+--------
        | F1 Tracker query      |                       |
        |======================>|                       |
        | F2 Tracker reply      |                       |
        |<======================|                       |
        | F3 ALTO client protocol query                 |
        |---------------------------------------------->|
        | F4 ALTO client protocol reply                 |
        |<----------------------------------------------|
        |                       |                       |

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

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

```
 .............................         .............................
 : Tracker                    :         : Peer                      :
 :   ______                   :         :                           :
 : +-______-+                 :         :                           :
 : |        |     +--------+ : P2P App. :  k good peers &  +------+ :
 : |   N    |     | ALTO-  | : Protocol :  n-k bad peers   | data | :
 : | known  |====>| biased |*******************************>| ex-  | :
 : | peers, |     | peer   | : transmit :                  | cha- | :
 : | M good |     | select | : n peer   :                  | nge  | :
 : +-______-+     +--------+ : IDs      :                  +------+ :
 :.....................^.....:          :...........................:
                       |
                       | ALTO
                       | client protocol
                     __|___
                   +-______-+
                   |        |
                   | ALTO   |
                   | server |
                   +-______-+
```

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

```
      Peer              Tracker w. 3PAC         ALTO Server
--------+--------      --------+--------      --------+--------
        | F1 Tracker query      |                      |
        |======================>|                      |
        |                       | F2 ALTO cli. p. query |
        |                       |---------------------->|
        |                       | F3 ALTO cli. p. reply |
        |                       |<----------------------|
        | F4 Tracker reply      |                      |
        |<======================|                      |
        |                       |                      |

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

Figure 4: Basic message sequence chart for third-party ALTO query

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 third-party discovery. 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 B. Contributors List and Acknowledgments

The initial version of this document was co-authored by Marco Tomsu <marco.tomsu@alcatel-lucent.com>.

Hannes Tschofenig provided the initial input to the U-NAPTR solution part. Hannes and Martin Thomson provided excellent feedback and input to the server discovery.

This memo borrows some text from [I-D.ietf-alto-server-discovery], as the 3pdisc 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/

Kilian Krause
University of Stuttgart Information Center
Allmandring 30
Stuttgart 70550
Germany

Email: schreibt@normalerweise.net
URI: http://www.rus.uni-stuttgart.de/nks/

Martin Stiemerling
NEC Laboratories Europe
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 113
Email: martin.stiemerling@neclab.eu
URI: http://ietf.stiemerling.org