ALTO Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 31, 2010

Z. Dulinski
Jagiellonian University
R. Stankiewicz
P. Cholda
P. Wydrych
AGH University of Science and Technology
B. Stiller
University of Zurich

June 29, 2010

Inter-ALTO communication protocol
draft-dulinski-alto-inter-alto-protocol-00

Abstract

The ALTO service provides the information, which can make communication between applications more efficient, especially in case of overlay applications. Such applications can use the information to perform better-than-random peer selection. The ALTO protocol conveys network information to applications. The protocol definition of this document extends the functionality of this ALTO service by introducing a standardized manner of communications between ALTO servers. A new inter-ALTO protocol is proposed, which enables the exchange of information between ALTO servers. The servers can coordinate actions and can introduce policies, which provide communication between applications localized in cooperating Autonomous Systems with a higher performance and a better cost efficiency.

Requirements Language

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 is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on December 31, 2010.

Copyright Notice

Table of Contents

1. Introduction

This document describes the communication protocol to be used between ALTO servers located in different autonomous systems (AS). The proposed inter-ALTO protocol extends the ALTO service [RFC5693] capabilities and provides additional information on remote peers, that is, peers located in other ASes. This information MAY be used by an ALTO server to perform advanced sorting/rating procedure of peers. The general idea is as follows:

1. A peer receives from a tracker a list of other peers - potential candidates to communicate with.

2. A peer uses the ALTO protocol [I-D.ietf-alto-protocol] to send the list of peers to its local ALTO server.

3. Local ALTO server obtains additional information on remote peers by communicating respective ALTO servers.

4. Using ISP specific policies and values of parameters associated with remote peers the local ALTO server performs sorting/rating procedure.

5. Sorted/rated list of peers is sent back to the peer.

The sorting/rating procedure is out of scope of this document. The inter-ALTO communication protocol that makes it possible to obtain extended information on remote peers is proposed.

To make the consideration more clear we distinguish local AS and remote ASes. Local AS is the one from which perspective we describe the communication. Local peers are located in the local AS and are served by a local ALTO server. On contrary, all other peers are located in remote ASes. Those peers are referred to as remote and are served by remote ALTO server. This basic terminology adheres to majority of considerations in this document.

2. Motivation

ALTO server optimization capabilities are limited by the fact that they use information available locally only. It can be shown that more information on remote peers, a routing path, or remote ISP preferences would be useful. The data from remote peer ASes will have a substantial significance for the management of overlay traffic (e.g. with respect to peer rating, sorting, or the choice of the best peers). The suggested approach to deliver these types of information is defined in the inter-ALTO communication protocol proposed.

In particular, the following key aspects motivate the proposal of an inter-ALTO protocol:

- Route asymmetry.
- Different types of business relations.
- Congestion avoidance.
- Proximity awareness (distance to the remote AS), e.g.:
  - number of inter-AS hops;
  - delay (RTT).
- Remote ISP preference.
- Coordination of ISPs' policies.

2.1. Route asymmetry

The communication between two ASes does not need to follow the same path in the upstream and downstream direction. It was shown that about 29% of paths between AS pairs in the Internet are fully symmetric, that is upstream and downstream traffic follows exactly the same path [Dulinski_ICC2010]. In 51% cases the number of inter-AS hops is different for the upstream and downstream direction. Additionally, in 50.5% of all path pairs a neighbor AS for upstream and downstream paths are different.

The ALTO server can obtain routing information locally (e.g. from BGP) and can determine the upstream path. Information about the downstream path is usually not easily available. Some additional routing information can be obtained from Looking Glass Servers, but not all ASes provide them. The inter-ALTO protocol supports the exchange of relevant information between ALTO servers. Especially, the downstream path can be reliably determined using the information

provided by remote ALTO server. In the light of route asymmetry in the Internet such information appears to be useful for a better optimization of a peer rating/sorting algorithm.

2.2. Different types of business relations

Two basic business relations between ISPs may be distinguished.

When two ISPs agree to exchange the traffic without any charge, such a relation is called peering. The inter-domain link between the respective ASes is also called a peering link. Usually, there is not charge if the difference between traffic volumes passing such a link in different directions does not exceed agreed limit.

Often one ISP serves as a network provider to another ISP (e.g. relation between tier 2 and tier 3 ISPs). In such a case one ISP (acting as a customer) has to pay the other ISP (acting as provider) for the traffic sent over the inter-AS link connecting them. The real monetary cost of the traffic volume exchanged on such a link depends on agreements between ISPs. In general, some links may be considered as cheaper or more expensive.

AS may be connected to many other ASes with various agreements. The cost of the inter-AS traffic transfer may differ depending on which neighbor AS the path passes. For this reason an ISP may prefer its own customers to exchange data with remote peers located in such ASes that the path to them passes cheaper links. The ALTO server may sort peers taking into account these criteria. To receive almost complete information on routing paths to different remote domains the information provided by remote ALTO server using inter-AS protocol can be helpful.

2.3. Congestion avoidance

A peer sorting procedure MAY also take into account the congestions on inter-AS links. An ISP can monitor queues on its inter-domain links and assign metrics indicating the buffer occupancy or bandwidth utilization. These metrics can express percentage use of buffers or bandwidth on a particular inter-AS link. If one inter-domain link is congested it is desirable to promote peers reachable through lightly loaded links. Again, information provided by the remote ALTO server would support such optimization.

2.4. Proximity awareness

For a set of reasons (e.g. the performance of an application) the ALTO server may suggest its customers to connect to remote peers located in its proximity. The simplest measure of proximity is the

number of inter-AS hops. Due to route asymmetry the number of hops may differ between upstream and downstream paths, as indicated above. Such information for the downstream path may be provided by the remote ALTO server. A more advanced metric of proximity can be found in the delay that can be approximated by exchanging messages between ALTO servers. The ALTO servers can be equipped with an application ping functionality which only operates between ALTO servers. By exchanging special packets prepared by the ALTO servers, these servers can estimate delay and packet loss.

2.5. Remote ISP preference

If two ISPs agree on a cooperation, the remote ALTO server MAY provide its preference parameters (remote preference parameters) indicating which peers are better from the point of view of the remote ISP. For instance, the AS in which the remote ALTO server is located may possess two subnets connected to the operator core network by distinct links. It may happen that a connection to one of the subnets is cheaper than the other. The remote operator may prefer connections through cheaper link, so peers located in the subnet transferring data via this cheaper link are preferred.

The remote preference parameter MAY be also used when a remote ISP wants to suggest peers which are conncected to the Internet through access links of higher capacity. This way, the remote ALTO server, without exposing the exact values of access link bandwidth, may indicate peers with higher throughput. The remote preference parameters have only local meaning, that is, their values are comparable for peers located in the same AS only.

If a remote ISP does not want to reveal numerical values of network parameters related to its peers (such information might be considered as confidential) the remote ALTO server may perform a sorting procedure and assign priority parameter to its peers. The sorting criteria MAY remain hidden for the requesting local ALTO server.

2.6. Coordination of ISPs' policies

Operators MAY coordinate their efforts in order to lower transfer costs on inter-domain links or improve transfer performance experienced by peers, namely coordinate peer sorting/rating strategies. This way operators may avoid contradictory strategies resulting in inefficiency of sorting/rating algorithms. Operators may agree to promote each others peers, e.g. by always placing peers serviced by the other party on the sorted/rated list amongst first 10 entries.

For example, it may happen that operator A wanting to decrease

traffic on one of its links discourages its own peers from communicating with peers located in operator B's domain. On the other hand, operator B would consider peers located in a domain of operator A as very attractive for its own peers. As a result, sorting/rating procedures performed by respective ALTO servers give contradictory results what may lower the effectiveness of these procedures. To avoid such a situation, the inter-ALTO protocol is needed.

Another example of a usefulness of coordination of policies is clustering of ASes. Recent studies have shown that locality promotion might be ineffective or even harmful if used in AS with small number of peers. A proposed solution is to create cluster of two or more ASes. Then ALTO servers serving different ASes in the cluster treat all peers located in the cluster as if they were in a single AS. In other words, from a point of view of locality promotion algorithm all peers located in the cluster are local, regardless of their home AS.

2.7. Sensitivity of topology information

The minimum information that the remote AS MUST provide to the local ALTO server via the inter-ALTO protocol are the number of inter-AS hops and the number of the local AS' neighbor in the downstream path (the full downstream AS_PATH MAY be not exchanged). Such information does not reveal any sensitive information neither on the ISP internal topology details nor remote AS connections with other ASes, but does provide basic and useful information for the local ALTO server.

If two ISPs or even more agree on the exchange of additional information, the protocol does allow for it.

3. Definitions

The inter-ALTO protocol enables communications between ALTO servers located in remote ASes. Communicating ALTO servers MAY belong to different ISPs. ISPs MAY decide to cooperate and exchange some set of parameters.

3.1. ALTO-ISP communities

The set of parameters exchanged between ALTO servers is classified as mandatory or optional. ALTO servers that agree on the exchange of a particular set of mandatory parameters form an ALTO-ISP community. These mandatory parameters MUST be exchanged always by ALTO servers belonging to a given ALTO-ISP community. By joining a particular ALTO-ISP community an ISP commits to be ready to send mandatory parameters to all other members of the community. A unique set of mandatory parameters constitutes the community.

The ALTO server MAY belong to many ALTO-ISP communities, depending on which set of mandatory parameters it is willing to exchange. An ISP MAY possess a few ALTO servers in order to separate the inter-ALTO traffic.

3.1.1. Mandatory parameters

The names of mandatory parameters and their meaning MUST be defined for each ALTO-ISP community.

All mandatory parameters defined for a given community MUST always be sent in a response to the request. A local ALTO server MAY use only a selection of received mandatory parameters for sorting peers or it MAY use none of them. Thus, the receipt of mandatory parameters does not oblige operators to use them for overlay management.

3.1.2. Optional parameters

The names of optional parameters and their meaning MUST be defined for each ALTO-ISP community.

Optional parameters MAY be exchanged on demand or on scheduled basis. These optional parameters MAY be requested by the local ALTO server, but the remote ALTO server MAY refuse to deliver them.

The remote ALTO server responding to the request MAY also send some unsolicited optional parameters. In this way a remote ALTO server suggests the local ALTO server additional criteria that MAY be used for sorting peers. For instance a remote ALTO server can send a remote preference parameter (described in Section 2.5) as an optional

parameter. The ALTO server to which the response is addressed MAY always ignore these parameters.

3.1.3. Parameter updates

It is assumed that for sorting/rating procedures ALTO servers mostly use parameters which are quite constant in time. ALTO servers SHOULD extensively cache received parameter values. Timers MAY be established for all cached parameters, and the update procedures MUST be decided during the parameter exchange.

Two update methods are defined: "push" and "pull".

The "pull" update method indicates that when a new value is expected, the local ALTO server sends a request with the name of the parameter (with a relevant peer list) for which the current value is required.

The "push" update method is used if a decision on when to send a new parameter value is left to the ALTO server responsible for this parameter. The ALTO server implements a timer. The value of the timer determines how often updates of this parameter value will be sent. If the value of a timer equals 0, it means that a remote site will send the current value, if a parameter value has changed (when a predefined event changing a value of parameter has happened). A value different from 0 defines update periodicity. A timer value MUST be defined, if the "push" update method has been chosen.

The update method MUST be negotiated between ALTO servers. An ALTO server sending a request for parameter values MAY suggest the update method ("pull" or "push" with a timer setting MAY be proposed). A decision about the update method is taken by the ALTO server sending a parameter value. Finally, an ALTO server that receives the parameter value and associated update method MAY accept this update setting or reject it. The "pull" method MUST be always accepted. If the "push" method is accepted a timer setting MUST also be accepted; no more negotiation of timer setting is allowed. If an ALTO server rejects "push" update method it means that it does not want to receive unsolicited updates. Then it may change the update method to "pull". It is done by requesting the same parameter again with the "pull" update method.

The "pull" method is considered as the lowest update requirement. A higher requirement is an event-based update (timer set to 0). The highest requirement is periodic update. If an update method was suggested by an ALTO server requesting a parameter value, the responding ALTO server MAY accept the proposed settings or MAY lower those setting requirements.

Communicating ALTO servers MAY change the update settings. The "push" method MAY be always changed to the "pull". The timer value MAY be changed to 0. Also, an ALTO server MAY resign from periodic updates anytime by sending a request with the related parameter name and the update parameter defined to the "pull" value.

3.2. GENERAL Community

The members of the GENERAL community MUST send information, which follows from the BGP AS_PATH attribute. There are two mandatory parameters:

- The autonomous system number of AS being a neighbor of a local AS with respect to the downstream path (from the remote-AS to the local-AS). The AS number is to be extracted from the AS_PATH attribute. This parameter is referred to as AS_neighbor.

- An integer value number, which expresses the distance between remote AS and local AS measured in the number of AS hops. The name for this parameter is AS_hops.

Sending the full AS_PATH information is OPTIONAL, since some operators may want to limit the proliferated information about the way their traffic comes out of their domain. If some operators agree to exchange the full AS_PATH, they MAY exchange it as a mandatory parameter in the frame of ISP defined community (see Section 3.3).

3.3. ISP defined communities

A group of operators MAY decide to create a set of mandatory parameters on their own. These ISPs define a community with a new set of mandatory parameters.

An ISP defined community MUST inherit mandatory and optional parameters from previously defined ALTO-ISP communities. They form an inheritance tree. A root of a community tree is always the GENERAL community, since belonging to the GENERAL community is REQUIRED. A new community MUST have one and only one ancestor community.

Each mandatory parameter of the ancestor community MUST be defined as mandatory parameter of the derived community. Each optional parameter of the ancestor community MAY be defined as mandatory or parameter of the derived community or MAY be defined as optional one. There is no limitation on the number and type of new mandatory and optional parameters within ISP-defined communities.

Any operator, which is a member of a given ISP-defined community,

MUST be a member of the ancestor community. Consequently, it MUST be a member of the GENERAL community.

3.4. Inter-ALTO server capability

An inter-ALTO server capability service MAY be provided by each ALTO server. This service running on a particular ALTO server MUST deliver the information on all communities supported by this server and also MUST describe the communities supported, i.e., it MUST provide names of the communities and the names of all the mandatory and optional parameters defined for each of the communities, the descriptions of all units used, and the relations between them.

This service MAY be used only by ALTO servers, the service MUST NOT be accessible by third party. The proper security measures MUST be undertaken in order to protect information storage and transfer. The service MAY also be used for proliferation of newly defined parameters in a particular ALTO server. Each ALTO server MAY limit the accessibility of some information for some ALTO servers (operators) through access lists. The detailed description of the inter-ALTO server capability service is out of scope of this document.

4. Protocol description

The local ALTO server can request parameters from a remote ALTO server. Depending on the community some parameters MAY or MUST be delivered by the remote ALTO servers. A remote ALTO server MUST respond to a request.

Inter-ALTO servers MUST use TCP to establish connections.

This section defines types and formats of request and response messages.

The Java Message Service [JMS] ObjectMessages are used to encode the messages.

4.1. Definitions of elements of request/response messages

The main elements that appear in request/response messages are as follows.

4.1.1. Community

Contains a name of a community. This element MUST be sent in any type of message. It can contain letters, digits, and the colon. The community name is case-insensitive:

community-name = 1*(ALPHA / DIGIT / ":")

4.1.2. Peer list

Contains individual or aggregated IP addresses of peers (e.g. 192.0.2.2/24). It is used either to:

- request parameters values for the peers on the peer list from a remote ALTO server, or
- to indicate the peers the provided parameters values apply to.

individual-peer = <address-family> <address>

aggregated-peers = <address-family> <address> <prefixlength>

peer-list = 1*(<individual-peer> / <aggregated-peers>)

4.1.3. List of parameters

For each parameter its name, and update method MUST be provided both in request and response messages. The names MUST contain only letters and digits, and are case-insensitive. Additionally, if an ALTO server sends a parameter value the meaning of this parameter MUST be specified. The parameter can be used for sorting or can have informational purpose. It takes values "descending" or "ascending" which indicates whether lower or higher values of the parameters should be preferred in case of sorting. The "info" value represents informational purpose.

parameter-name = 1*(ALPHA / DIGIT)

parameter-meaning = <meaning-info> / <meaning-asc> / <meaning-desc>

An update method MUST be established for all exchanged parameters since most recent parameter values are necessary for proper peer sorting/rating procedure. "Pull" or "push" update method may be chosen. If "push" method is used a timer value MUST be specified.

parameter-update-method = <method-pull> / (<method-push> <timer>)

4.1.4. Suggested parameter significance sequence

An ALTO server sending parameter values MAY suggest the other party the sequence in which the parameters should be taken into account for peer sorting/ranking procedure. In other words, this indicates the sequence of the parameter importance from the point of view of ALTO server sending the parameter values. For this purpose, the ALTO server MAY send ordered list of parameter names. An ALTO server receiving parameter values MAY use this sequence exactly as proposed, partially, or MAY completely ignore it, and thus decide which parameters take into account and in which order on its own.

significance-sequence = 1*<parameter-name>

4.2. Requests

Two types of request are defined:

- an EXTENDED REQUEST and
- a BASIC REQUEST.

For each request a response MUST be sent. Both types of requests are sent within a community. The selected community MUST be specified in each type of request.

4.2.1. EXTENDED REQUEST

The EXTENDED REQUEST is used to ask a remote ALTO server for all mandatory parameters defined within a frame of a community specified in the request. Some optional parameters MAY be requested. It is expected that the remote ALTO server will be interested in parameters related to local peers, that is located at requesting party&s AS, and would request them in the near future. To reduce the number of exchanged messages, a local ALTO server places parameters values for local peers in the EXTENDED REQUEST and sends them to remote ALTO server although they are unsolicited.

There are two main parts of the EXTENDED REQUEST message: "local parameters" and "remote parameters".

All parameters describing local peers are placed in the "local parameters" section of the request. A local ALTO server MUST sent values of all its mandatory parameters. Additionally a local ALTO server MAY send values of optional parameters describing those local peers. For all local parameters an update method MUST be established. A local ALTO server MAY suggest a parameter significance sequence by sending ordered list of local parameter names.

The "remote parameters" part of the message is used to specify the list of remote peers for which a local ALTO server request values of mandatory parameters. Values of all mandatory parameters defined for a given community MUST be requested in the EXTENDED REQUEST. Additionally, a local ALTO server MAY request a remote ALTO server for some optional parameters. The format of messages is organized in such a way that parameters names together with attributes precede a peer list which relates to them. Such groups MAY appear in a message many times.

If a remote ALTO server agrees to respond to the EXTENDED REQUEST, it MUST respond with all mandatory parameters defined for a specified community. A remote ALTO server MAY refuse to respond in the frame of a community specified in a request.

The format of an EXTENDED REQUEST message is defined as outlined below:

```
extended-request  = <request-type-extended> <community-name>
                    <local-parameters> <remote-parameters>

local-parameters  = <mandatory-local-parameters>
                    [<optional-local-parameters>]
                    [<significance-sequence>]

remote-parameters = <mandatory-remote-parameters>
                    [<optional-remote-parameters>]

mandatory-local-parameters  = 1*(1*<local-parameter> <peer-list>)

optional-local-parameters   = 1*(1*<local-parameter> <peer-list>)

mandatory-remote-parameters = 1*(1*<remote-parameter> <peer-list>)

optional-remote-parameters  = 1*(1*<remote-parameter> <peer-list>)

local-parameter  = <parameter-name> <unit> <parameter-value>
                   <parameter-meaning> <parameter-update-method>

remote-parameter = <parameter-name> <parameter-update-method>
```

4.2.2. BASIC REQUEST

The BASIC REQUEST message MAY be used by the local ALTO server to request for any subset of mandatory parameters defined for a specified community as well as optional parameters. Any mandatory or optional parameter MAY be requested. Specifically, the basic request MAY be used for requesting a single parameter (mandatory or optional).

In this request, the requesting party does not send parameters of local peers. The message consists only of "remote parameters" part. It contains the list of requested parameters, both mandatory and optional ones, and the list of remote peers the parameters are requested for. Similarly to the EXTENDED REQUEST, an update method for requested parameters MUST be suggested by requesting party. The negotiation of the update method is described in Section 3.1.3.

The format of a BASIC REQUEST is defined as outlined below:

```
basic-request = <request-type-basic> <community-name>
                <remote-parameters>
```

4.2.3. Recommended usage of requests

A few types of applications may be distinguished: the ones which perform uploading and downloading, and the other which only download. For the simultaneously uploading and downloading applications the EXTENDED REQUEST is RECOMMENDED. It is expected that the ALTO server from the peer AS will request parameters, at least obligatory parameters. In order to limit message transfer the local ALTO server sends the values of all local obligatory parameters in advance in the EXTENDED REQUEST. For downloading only applications, the BASIC request is to be used. Note that applications which only upload content may use only information available in the local AS and the local ALTO server does not need to communicate with the remote ALTO server.

The more advanced sorting/rating procedures may require information from the remote AS for all types of applications.

4.3. Responses

A remote ALTO server SHOULD always send a response to the requests received.

A remote ALTO server can receive requests for optional parameters. It depends on the operator's policy possessing an ALTO server, which optional parameters values MAY be sent in a response. If a particular optional parameter is not supposed to be sent, a responding ALTO server does not place the parameter in a response.

4.3.1. REFUSE RESPONSE

The REFUSE RESPONSE is sent when a responding ALTO server does not want to communicate with the requesting ALTO server within the indicated community. In other words, a responding ALTO server informs the requesting party that in the frame of the community specified there will be no communication between them. Operators SHOULD regulate these actions through policies (e.g. access lists).

REFUSE RESPONSE message MUST NOT be sent if the request is within the GENERAL community.

If an ALTO server requests some mandatory parameters, which are out of the scope of the community specified in that request, it SHOULD be treated as an attempt to swindle data. A responding party SHOULD send the REFUSE RESPONSE message.

refuse-response = <response-type-refuse> <community-name>

4.3.2. ERROR RESPONSE

The ERROR RESPONSE message MAY be used when an unrecognized parameter name has been received in a request. Having received a request with wrong names, a responding ALTO server MAY optionally send a set of unrecognized parameter names:

```
error-response = <response-type-error> <community-name>
                 <wrong-parameter-names>

wrong-parameter-names = 1*<parameter-name>
```

If other errors appear they SHOULD be processed by lower level protocols.

4.3.3. NORMAL RESPONSE

An ALTO server uses NORMAL RESPONSE message in order to respond to both types of requests. NORMAL RESPONSE message is also used by ALTO servers for generating parameter updates, both periodic and event-based. When an ALTO server responds to EXTENDED REQUEST, it MUST send values for all mandatory parameters defined for the community specified in the request.

When an ALTO server responds to BASIC REQUEST it MUST sent values only for those mandatory parameters which have been requested (all or a subset of mandatory parameters defined for community specified in the request).

If an ALTO server does not want to communicate with a requesting ALTO server within the community specified in the request it MUST NOT send any parameters and then MUST send RESPONSE REFUSE.

If optional parameters were requested (in either BASIC or EXTENDED REQUEST) the responding ALTO server MAY sent their values in NORMAL RESPONSE. Sending values of optional parameters is OPTIONAL.

An ALTO server sending NORMAL RESPONSE MAY also send additional, not requested optional parameters for the peers specified in the request. In this way, the responding ALTO server may suggest additional parameters it wants to be used by requesting ALTO server for a sorting/rating procedure. Together with a parameter name, its value MUST be sent and the update method MUST be specified.

The ALTO server sending the NORMAL RESPONSE MAY suggest a parameter significance sequence by sending ordered list of the parameter names.

The proposed format of NORMAL RESPONSE is defined as follows:

```
normal-response = <response-type-normal> <community-name>
                  <mandatory-local-parameters>
                  [<optional-local-parameters>]
                  [<non-requested-optional-local-parameters>]
                  [<significance-sequence>]

non-requested-optional-local-parameters =
                  1*(1*<local-parameter> <peer-list>)
```

4.4. Message exchange patterns

This section presents three sample scenarios showing the idea of inter-ALTO protocol and message exchange. In Figures 1, 2, and 3, the local ALTO server communicates only one remote ALTO server. This is done for readability. A local ALTO server SHOULD asynchronously communicate as much remote ALTO servers as it is needed.

4.4.1. Successful communication within a given community

1. The local ALTO server receives a peer list from a local peer (amongst the peers on this list there are the peers located in the remote AS).

2. For each peer from the list, the local ALTO server searches in the cache for parameters necessary for sorting/rating. If the necessary parameters have been retrieved for a particular peer, these parameters are assigned to that peer. If the necessary parameters are absent in the cache, the local ALTO server discovers the address of the remote ALTO server (the peer address is the input parameter for discovery procedure).

3. The local ALTO server sends the request (BASIC or EXTENDED) to the remote ALTO server, specifying a community and required parameters.

4. The remote ALTO server sends the NORMAL RESPONSE to the local ALTO server. The local ALTO server assigns parameters to the peers according to the received response. The received parameters SHOULD be cached.

5. After parameter assignment to all peers from the list, the peer sorting/rating procedure is performed.

6. The sorted list is sent to the local peer.

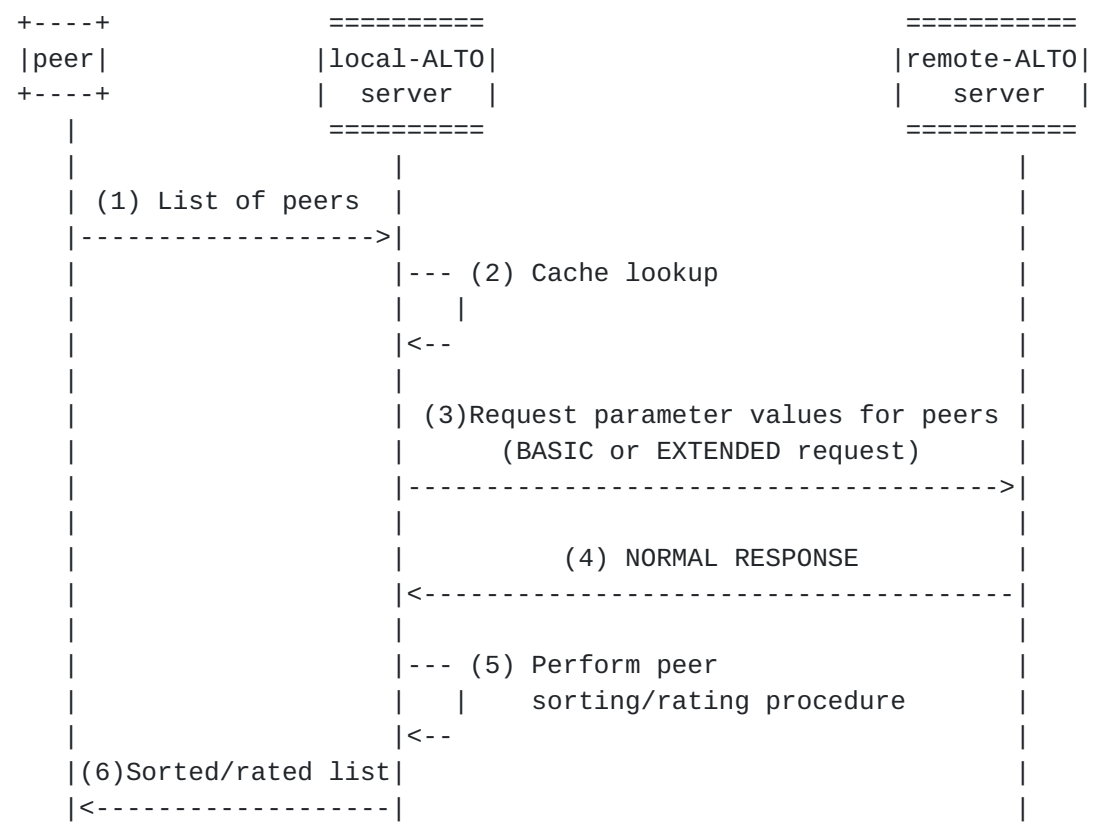

Figure 1: Successful message exchange.

4.4.2. Rejected communication within the given community

If a remote ALTO server rejects communication within the frame of a specified community, the local ALTO server MAY lower community requirements and send the request again.

1. The local ALTO server receives a peer list from a local peer (amongst the peers on this list there are the peers located in the remote AS).

2. For each peer from the list, the local ALTO server searches in the cache for parameters necessary for sorting/rating. If the necessary parameters have been retrieved for a particular peer, these parameters are assigned to that peer. If the necessary parameters are absent in the cache, the local ALTO server discovers the address of the remote ALTO server (the peer address is the input parameter for discovery procedure).

3. The local ALTO server sends the request (BASIC or EXTENDED) to the remote ALTO server, specifying a community and required parameters.

4. The remote ALTO server sends the REJECT RESPONSE to the local ALTO server.

5. The local ALTO server sends the request (BASIC or EXTENDED) to the remote ALTO server, specifying a community with smaller set of mandatory parameters.

6. The remote ALTO server sends the NORMAL RESPONSE to the local ALTO server. The local ALTO server assigns parameters to the peers according to the received response. The received parameters are cached.

7. After parameter assignment to all peers from the list, the peer sorting/rating procedure is performed.

8. The sorted list is sent to the local peer.

Steps 4 and 5 MAY be repeated as many times as it is needed.

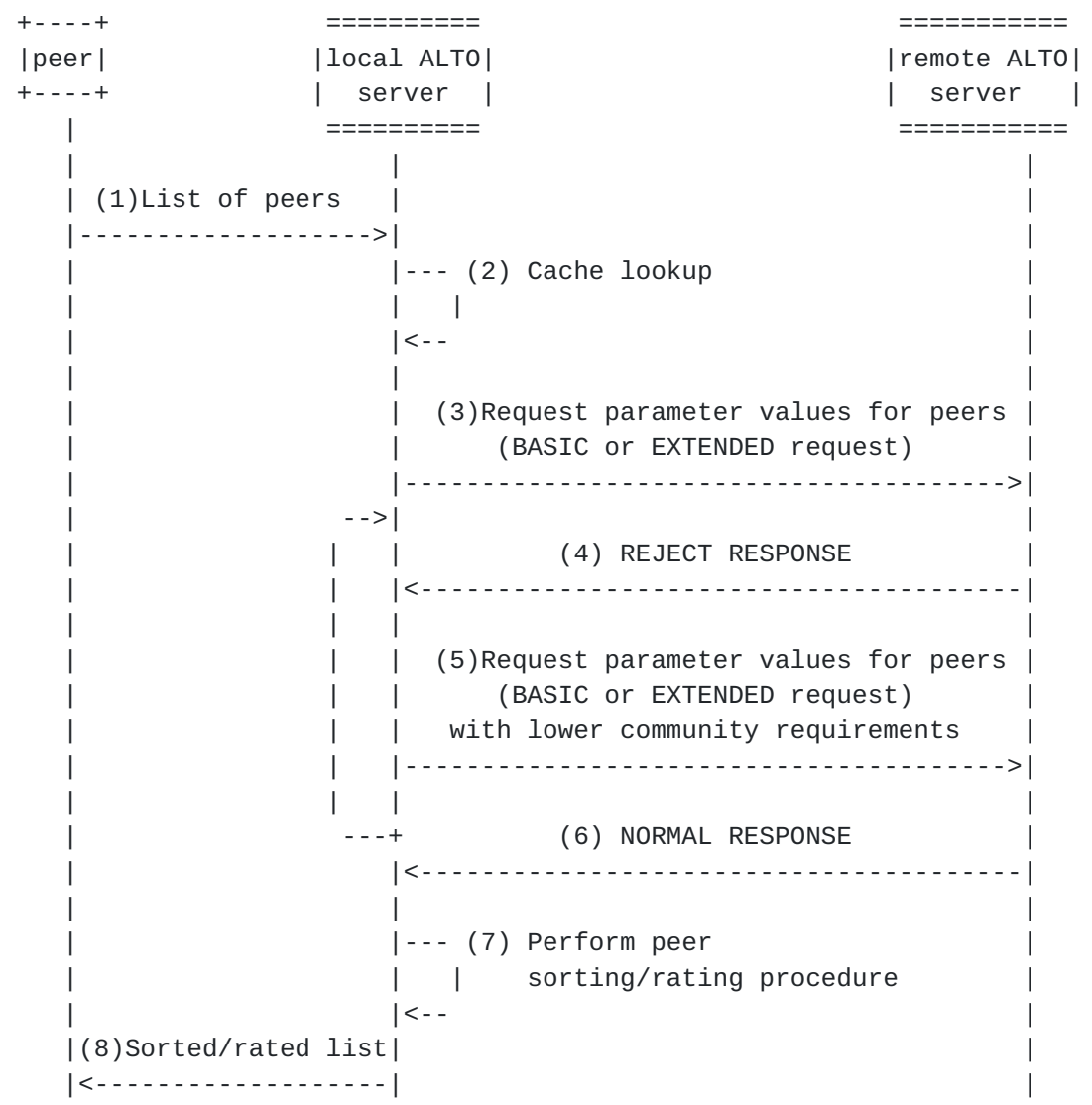

Figure 2: Message exchange in the case of rejected communication within the given community.

4.4.3. Handling wrong parameter names in the request

1. The local ALTO server receives a peer list from a local peer (amongst the peers on this list there are the peers located in the remote AS).

2. For each peer from the list, the local ALTO server searches in the cache for parameters necessary for sorting/rating. If the necessary parameters have been retrieved for a particular peer, these parameters are assigned to that peer. If the necessary parameters are absent in the cache, the local ALTO server discovers the address of the remote ALTO server (the peer address is the input parameter for discovery procedure).

3. The local ALTO server sends the request (BASIC or EXTENDED) to the remote ALTO server, specifying a community and required parameters.

4. The remote ALTO server discovers wrong parameter names, it sends the ERROR RESPONSE to the local ALTO server. The erroneous parameters are specified in the response.

5. The local ALTO server performs sorting/rating with incomplete knowledge.

6. The sorted list is sent to the local peer.

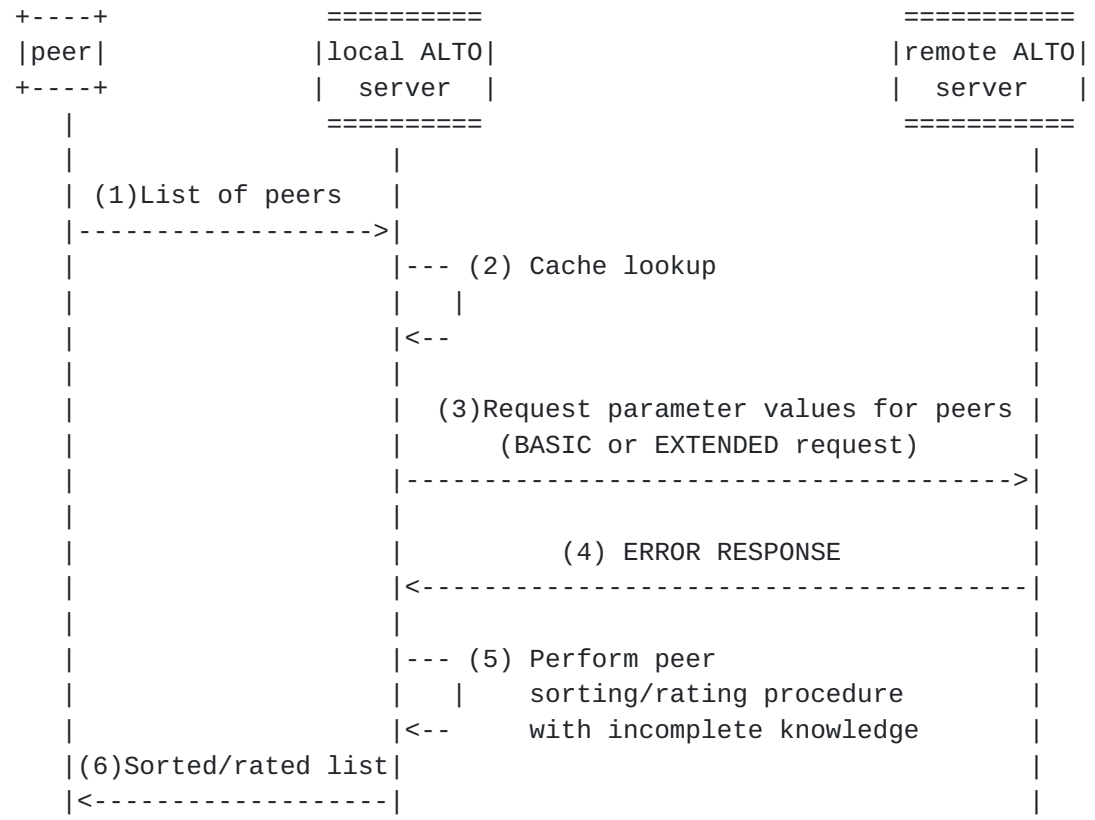

Figure 3: Message exchange in the case of sending wrong parameter names.

5. Inter-ALTO server discovery

The local ALTO server needs to know the IP address of the remote ALTO server in order to contact with this remote server. A service which enables an ALTO server discovery has to be proposed.

The main assumptions for this service are described below.

The local ALTO server receives peer list from a peer. This list contains IP addresses of peers. For some remote peers the IP address of remote ALTO server may be unknown. The local ALTO server, using peer IP addresses, has to able to establish the addresses of the remote ALTO servers.

In order to determine IP address of a remote ALTO server quickly and to limit the network load, a database of IP addresses of all ALTO servers should be stored locally in each ALTO server. All database search are performed locally unless the remote ALTO server for a given peer is unknown. The database stores network prefixes announced by BGP together with the IP address of an ALTO server serving these prefixes. The database may also contain information about the AS number and served communities to which a particular ALTO server belongs.

Two approaches to the remote ALTO server discovery are proposed: centralized and distributed.

The centralized approach assumes the existence of so called info-ALTO servers that are supposed to be managed by a trusted organization, which ensures a proper level of security and confidentiality. The organization managing info-ALTO servers is responsible for registering ALTO servers and for the verification of ISPs that want to join a selected ALTO-ISP community. Info-ALTO servers store necessary information for ALTO servers' localization and communication. They can be found by DNS.

In a decentralized approach it is assumed that ALTO servers will establish adjacency with some other ALTO servers and will exchange databases containing IP addresses of ALTO servers with them. A new ALTO server, which requires to exchange information with other ALTO servers, MUST communicate with any ALTO server, which is using this service. It must establish adjacency with at least one existing ALTO server in this context (this server is called old ALTO server). Each old ALTO servers stores a database of addresses of all other old ALTO servers. The new ALTO server and the old ALTO server perform an authorization and an authentication procedure. In the next step the new ALTO server downloads the database from the old ALTO server.

In case of a database change, the old ALTO server is responsible for delivering updates to the new ALTO server. In an update, only changes in the database MUST be delivered, and not the entire updated database.

Detailed specification of inter-ALTO server discovery procedures is out of the scope of this document. It is left for a separate draft.

6. Reliability considerations

Reliability, that is fault-tolerance to failures, is a basic feature that SHOULD be provided to the inter-ALTO protocol. Lack of functionality in the case of inter-ALTO protocol may lead to two important problems: losses related to suboptimal flow of traffic caused by the application layer routing and decrease of the credibility of the operator in the inter-AS environment.

A successful operation of inter-ALTO protocol involves the proper work of a local ALTO server that decides that a new inter-ALTO query is necessary to be sent to a remote ALTO server with the usage of global Internet. To find this remote ALTO server a usage of inter-ALTO server discovery might be necessary. Therefore, the reliability of the inter-ALTO protocol is dependent on four factors: reliability of a local ALTO server, reliability of a remote ALTO server, reliability of underlying IP networks, and reliability of the inter-ALTO server discovery. They are described below in following sections.

The reliability of the whole protocol operation is dependent serially on the four enumerated factors, that means a failure of at least one of them makes the whole operation of the inter-ALTO protocol for a pair of ALTO servers in different ASes impossible. Thus, having the assessment of the reliability metrics, for instance in terms of the steady-state availability, of all four components, it is possible to assess the resulting reliability (e.g. as the product of the availabilities).

The reliability assessment is necessary to find the weak points of the system and to predict the output reliability metric to decide if it meets the operator's requirements. As a result, it is possible to improve the fault-tolerance of the components due to which the reliability is below expectations.

6.1. Reliability of a local ALTO server

Reliability of a local ALTO server is determined by elements which this server is built of. As it is a typical networking computing system, it depends on two kinds of building blocks, that is: software and hardware. Both serving the storage, computational logic and networking part.

From all reliability features impacting the operation of the inter-ALTO protocol, the ones related to a local ALTO server are most controllable by an operator. Therefore, an operator SHOULD take care of this component most and maximize the related reliability metrics for it. There are three main options for protecting the local ALTO

against negative impact of failures:

- introduction of fast restarting procedures to enable effective software reaction to errors that otherwise might cause breakdown of the local ALTO server operation;

- introduction of redundant hardware elements (with supporting software) to provide backup(s) in case the basic elements fail;

- partitioning of operation of different functionalities of the system to provide partial operation when some elements fail.

Both latter options are described below.

6.1.1. Redundancy of elements

Redundancy, that is introduction of additional elements that will replace the faulty elements, is a basic option for improving reliability of local ALTO server. It is possible to introduce redundancy at the level of a single server (e.g. by adding internally additional CPUs, storage or memory facilities etc.) or append the system with a complete alternative server. Three options for cold, warm and hot backup are available. Except for having a basic fault-tolerance consisting in subsistence of the functionality, it is necessary to have a system that makes the working server and its backup consistent. Thus, warm or hot backup is advised in order to update the storage and memory contents online. Then, it is possible that after a failure, the backup server does not start to work with empty caching information and there is not temporary performance degradation (caused by necessity to find inter-AS data for previously known prefixes) nor necessity to use the inter-ALTO service discovery in relation to previously known remote ALTO servers.

The fact that the local ALTO server uses backup should be masked from the viewpoint of the remote ALTO server in order not to make the protocol operation too complex, e.g. to omit necessity for dealing with changed IP addresses. Also the recovery time should be diminished as much as possible in order to avoid the switching to be noticed.

6.1.2. Partitioning of functionalities

Partitioning of functionalities of an local ALTO server is an option not dependent on redundancy and can be combined with. It consists in using separate entities (obtained by virtualization of a software/ hardware system or by implementation of additional facilities) for serving group of functions that can be logically separated in order to decrease the impact of failures on the whole operation. The

following options are useful:

- partitioning of functionalities supporting (1) transfer of information from/to local peers, and, (2) communications with remote ALTO servers: due to this option it is possible to sustain the inter-ALTO operation even in the situation when the operator cannot provide the ALTO functionality in its AS (e.g. due to denial of service attack initiated in its domain);

- partitioning of functionalities supporting different communities.

As an additional option it is possible to consider the situation when element supporting a separate functionality is a backup for element supporting another functionality.

6.2. Reliability of a remote ALTO server

The reliability of a remote ALTO server is independent of the local operator. However, the operation of the local ALTO service is dependent on the reliability of remote ALTO servers as they are used to gain information interesting for sorting/rating. An operator can assess the reliability of remote ALTO server as similar to the one provided to its local ALTO server as those components serve analogous functions. From the viewpoint of the system operation it is necessary that the remote AS provides fault-tolerance to its remote ALTO server in a way that the failures are not visible to the local ALTO server.

6.3. Reliability of underlying IP networks

The reliability of underlying IP networks is the component to some extent independent of two communicating parties. The communication chain typically passes the local AS, different ASes in the Internet independent of the communicating parties, and the remote AS. To improve the fault tolerance of the networking communications the connections used for supporting transfer of inter-ALTO messages SHOULD use recovery techniques adequate for providing fault-tolerance to connections against network failures (e.g. 1:1 or 1+1 protections or re-routing procedures). Additionally, the inter-ALTO messages MUST be supported by an underlying protocol providing retransmission of lost data and information protection at the coding level (e.g. by FEC).

6.4. Reliability of the inter-ALTO server discovery

The reliability of an inter-ALTO server discovery is independent of the local operator. However, the operation of the local ALTO service is dependent on the reliability of inter-ALTO server discovery as it

is used to find out addresses of remote ALTO servers of ASes not contacted before. Contrary to the three other components described above, there are cases when the reliability of this component does not influence the overall operation. Hence a negative impact on the whole inter-ALTO protocol operation in a single AS is not very strong. The exception is related to the situations when the discovery functionality is down when the local ALTO server has empty caches and needs to intensively use the inter-ALTO server discover functionality.

When the centralized approach to inter-ALTO server discovery is adopted, it is maintained by a specialized institution and the reliability of this component will be high. In the case of the decentralized approach the unavailability of the service means that many remote ALTO servers are down and the operation of the whole inter-ALTO protocol is jeopardized.

From the viewpoint of the system operation it is necessary that the discovery functionality is fault-tolerant in a way that the failures of it are not visible to local ALTO servers.

7. Scalability considerations

Scalability is a significant requirement of inter-ALTO protocol. The main threat related to it concerns two situations:

- too large number of queries to be effectively served generated by local peers resulting in necessity for a burdening communications with remote ALTO protocols,

- too large number of queries to be effectively served due to a significant load related to necessity to reply to many simultaneous requests from remote ALTO servers.

While the former problem is easier to solve, as the local ALTO server can simply temporarily cease to response to local peers or queue them (and increases delays in serving the sorting/rating requests), the latter situation is more challenging, as the response to inter-ALTO messages is mandatory. Then, the response in a relatively short time is necessary. In case of breaking this rule, the local ALTO server is erroneously perceived as either faulty or not conforming to the recommendations of this draft.

8. IANA Considerations

The IANA has registered "inter-alto" as TCP port number TDB1.

9. Security Considerations

ALTO server possesses information about the network topology and it MAY share this information with other servers through the inter-ALTO communication. Potential intruders can make an attempt to eavesdrop transmission in order to gain confidential information, i.e., by spoofing an ALTO server. One of the most dangerous attack could be the data modification during transmission between ALTO servers. This type of interference can result in wrong management of the network.

During the considerations about security of inter-ALTO protocol the crucial issue is proper balance between security and performance. Too many protection mechanisms implemented in the protocol can degrade efficiency. Nevertheless, the inter-ALTO protocol SHOULD provide basic security services at least. Below, some crucial security services and general solutions was presented.

9.1. Authorization

Authorization (access control) service is based on desired behaviors of ALTO servers. Each entity SHOULD belong to some group of privileges and have specific roles and permissions. First of all, the membership of the communities influences ALTO server permissions. An attempt of unauthorized access to resources should cause the RESPONSE REFUSE message.

9.2. Authentication

Authentication service SHOULD be applied by ALTO server, because other server must be sure that are connecting to proper entity. One of the best way to achieve this aim is certificate provided by server. By means of certificate, which can be validating by trusted entity, the ALTO server is able to confirm the identity.

9.3. Data confidentiality

Data encryption ensures secure transfer of sensitive information. If server has his own certificate, encryption can be realized by means of asymmetric ciphers. The main advantage of this solution is the usage of the public keys for the message encryption, which eliminates the problem of the secret key distribution or agreement. To ensure better performance, the messages could be encrypted by symmetric cipher but session keys could be encrypted by asymmetric ciphers.

9.4. Data integrity

Data integrity assures that during communication between ALTO servers, data must not change imperceptibly. This requirement is met

by means of one-way hash functions. Content of the messages SHOULD be protected by data integrity assurance to avoid any modifications.

9.5. Availability

The availability is assured by good quality of system design and implementation process as well as redundancy of system resources. Such attacks as Denial of Service (DoS) or Distributed DoS (DDoS) attacks can be performed to prevent or inhibit normal use of ALTO server. These attacks are performed by flooding the server with unwanted traffic, i.e. false inter-ALTO requests. To ensure the protection of ALTO server, it MAY be secured by external devices, such as Intrusion Detection/Prevention Systems (IDS/IPS).

10. Contributors

The people listed here should be viewed as co-authors of the document. Due to the limit of 5 authors per draft the co-authors were moved to the contributors section at this point.

- Marcin Niemiec (AGH University of Science and Technology)

- Miroslaw Kantor (AGH University of Science and Technology)

11. Acknowledgements

This work has been partially supported by the EU through the ICT FP7 Project SmoothIT (FP7-2007-ICT-216259).

The authors would like to thank Fabio Hecht from University of Zurich for his helpful comments.

12. References

12.1. Normative References

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

12.2. Informative References

[Dulinski_ICC2010]
Dulinski, Z., Kantor, M., Krzysztofek, W., Stankiewicz, R., and P. Cholda, "Optimal Choice of Peers based on BGP Information", Proceedings of 2010 IEEE International Conference on Communications (ICC), May 2010.

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-04 (work in progress), May 2010.

[JMS] "Java Message Service (JMS)", <http://java.sun.com/products/jms/>.

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

Authors' Addresses

Zbigniew Dulinski
Jagiellonian University
Reymonta 4
Krakow 30-059
Poland

Phone: +48 12 663 5571
Fax: +48 12 633 4079
Email: dulinski@th.if.uj.edu.pl

Rafal Stankiewicz
AGH University of Science and Technology
al. Mickiewicza 30
Krakow 30-059
Poland

Phone: +48 12 617 4036
Fax: +48 12 634 2372
Email: rstankie@agh.edu.pl

Piotr Cholda
AGH University of Science and Technology
al. Mickiewicza 30
Krakow 30-059
Poland

Phone: +48 12 617 4036
Fax: +48 12 634 2372
Email: piotr.cholda@agh.edu.pl

Piotr Wydrych
AGH University of Science and Technology
al. Mickiewicza 30
Krakow 30-059
Poland

Phone: +48 12 617 3805
Fax: +48 12 634 2372
Email: piotr.wydrych@agh.edu.pl

Burkhard Stiller
University of Zurich
Binzmuhlestrasse 14
Zurich CH-8050
Switzerland

Phone: +41 44 635 67 10
Fax: +41 44 635 68 09
Email: stiller@ifi.uzh.ch