Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: August 25, 2012

S. Madhavan
R. Nandiraju
Huawei Technologies

February 22, 2012

ALTO Caching and Subscription
draft-alto-caching-subscription-00

Abstract

The specification of the ALTO protocol uses map based approaches assuming that the information provided is static for a longer period of time. But in some cases network operators reallocate IP subnets from time to time which in turn changes the mapping partitions[I-D.ietf-alto-deployments]. Since the ALTO clients are unaware of the map information changes, clients need to query the servers for every service request and many such requests are redundant because the information was not changed. The purpose of this memo is to provide two mechanisms which will help the ALTO clients to be informed of the map information changes. a) ALTO clients cache the map information and the servers provide the expiration time for invalidation. b) ALTO clients can subscribe for event notifications from the server

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 August 25, 2012.

Table of Contents

1. Introduction 3
2. Problem Statement 3
3. Protocol Structure 4
4. Overall Operation 4
5. Definitions 5
6. ALTO Types 6
6.1. EndpointAddrGroup 6
6.2. NetworkMapData 6
6.3. ContactAddr 6
7. ALTO caching and invalidation 6
7.1. Server behavior 7
7.2. Client behavior 7
7.3. Examples 7
7.3.1. Network Map with Expiry Example 8
7.3.2. Filtered Network Map with Expiry Example 9
8. Subscription and Notification Service 9
8.1. Requesting a Subscription 10
8.1.1. Input Parameters 10
8.2. Refreshing of Subscriptions 11
8.3. Notifier Subscription Handling 11
8.3.1. Response 11
8.4. Notifier Notification behavior 12
8.4.1. Input Parameters 12
8.5. Subscriber Notification handling 13
8.6. Unsubscribing 13
8.7. Example 13
9. Detecting support for Subscription Service 16
9.1. Example 16
10. Transport Considerations 18
11. Acknowledgements 18
12. IANA Considerations 18
12.1. application/alto-* Media Types 18
13. Security Considerations 18
14. References 18
14.1. Normative References 18
14.2. Informative References 18
Authors' Addresses 19

1. Introduction

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource [RFC5693]. The requirements for ALTO are itemized in I-D.ietf-alto-reqs. ALTO is realized by a client-server protocol.ALTO clients send queries to ALTO servers, in order to solicit guidance.

ALTO clients needs to query the ALTO server for cost map and network map information to select one or several hosts from a set of candidates. Network Maps may be stable for a longer time but changes whenever the service provider reallocates IP subnets. [I-D.ietf-alto-deployments]

This memo defines a mechanism for ALTO clients to cache the map information and for the servers to provide the expiration time for invalidation at subset level or map level. In addition the ALTO clients can subscribe to the ALTO server to be informed of a map information change. The server MAY send only the changed map information even if clients subscribe for complete information.

2. Problem Statement

The ALTO protocol uses HTTP for discovering available Information resources at an ALTO server and retreiving information resources. The protocol defines Map-based and Non-map based approaches for retreiving information resources.Map-based approaches are chosen as they lower the signaling load on the server, as the maps have only to be retrieved if they are changed.

HTTP protocol already defines mechanisms for caching the content. There are many limitations in using those mechanisms. HTTP based caches store the responses only for HTTP GET requests. ALTO queries can be GET or POST(in case of filtered map) requests. In the case of POST requests, HTTP caching mechanism cannot be used because responses for POST are generally not cached.

The validity of the information in the map need not be same for all the elements in the map and there is currently no mechanism(in HTTP also) to convey this information. This mechanism has to be incorporated into the protocol itself.

ISPs reallocate IPv4 subnets within their infrastructure from time to time, partly to ensure the efficient usage of IPv4 addresses. The frequency of these "renumbering events" depend on the growth in

number of subscribers and the availability of address space within the ISP.

3. Protocol Structure

The ALTO Protocol uses a simple extensible framework to convey network information. In the general framework, the ALTO protocol will convey properties on both Network Locations and the paths between Network Locations.[I-D.ietf-alto-protocol].

The Map filtering and endpoint cost services can be extended to include the map level and subset level expiration time. This memo also extends the ALTO information services to include the subscription and notification service. Any node can transmit a notification through HTTP to any other node for the changes in subscribed events. The framework also provides mechanisms for subscriptions to be done for full maps or subset of the maps available in the Map service.

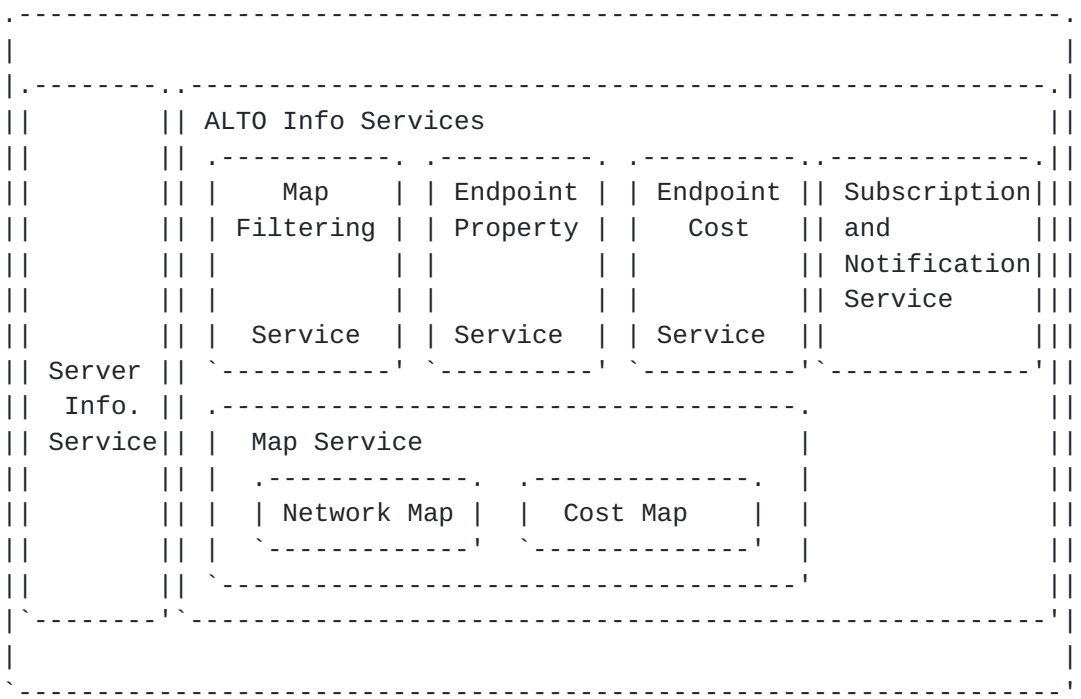


Figure 1: ALTO Protocol Structure

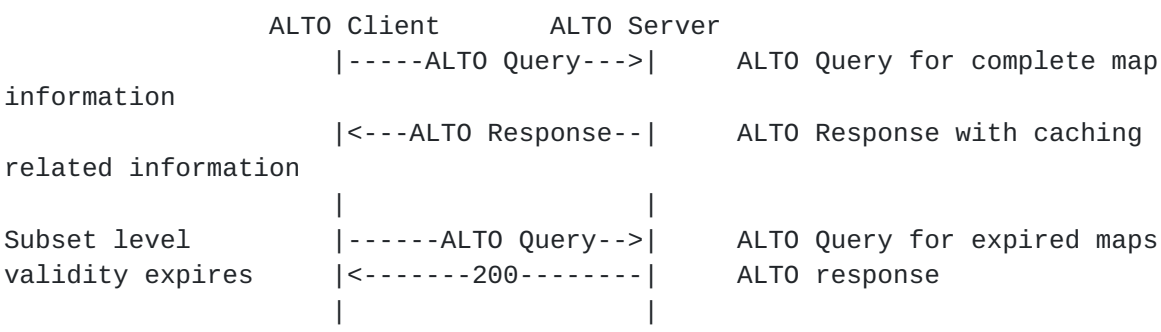
4. Overall Operation

The overall operation is based on the ALTO client caching the map information. The ALTO server sends the map information providing

details of full or subset level expiration. Clients can subcsribe for map information changes. ALTO server sends notifcations for the map information based on the subscription

ALTO server can choose to use HTTP based caching mechanisms or ALTO caching mechanism (defined in this document) based on the request.

A typical message flow would be:



The requests and responses are encoded with JSON. Subscriptions can be expired and MUST be refreshed using subscription requests.

Figure 2: Message flow

5. Definitions

ALTO cache: Any logical entity which caches the ALTO responses for the purpose of reducing repeated requests by the alto client to the alto server. The ALTO cache MAY be co-located with the ALTO client or it can be a separate node

Event:Any change in a resource that can trigger a notification

Subscription:An established relationship in which a resource has indicated interest in certain event

Subscriber:A subscriber is an ALTO node that initiates a subscription with a subscription server

Notification:Notification is the mechanism by which notifier nodes sends changed events information to the subsctiber

Notifier:A notifier is an ALTO node which generates Notification messages for the purpose of notifying subscribers of the state of a resource.

6. ALTO Types

This section details the format for particular data values used for ALTO caching and the Subscription framework. Base types defined by I-D.ietf-alto-protocol are used by this memo with some extensions listed below.

6.1. EndpointAddrGroup

EndpointAddrGroup JSON object is extended as below

```
object {
     JSONString expires;
     EndpointPrefix [AddressType]<0..*>;
     ...
   } EndpointAddrGroup;
```

6.2. NetworkMapData

NetworkMapData JSON object is extended as below

```
 object {
     JSONString expires;
     EndpointAddrGroup [pidname]<0..*>;
     ...
   } NetworkMapData;
```

6.3. ContactAddr

The ALTO subscription framework requires the client nodes to provide the contact information which includes the transport tuple details so that server can send notification. These connections can be persistent or not.

ContactAddr is defined as:

```
object {
     EndpointAddr addr;
     JSONInteger port;
   } ContactAddr;
```

7. ALTO caching and invalidation

ALTO servers specify the expiration time of the complete map or the subset level maps in the response. Expiration time format is an absolute date and time with GMT as reference. ISPs MAY use any mechanism to determine the expiration time. This is out of the scope of this specification.

7.1. Server behavior

ALTO server MAY use HTTP based or ALTO caching mechanism to convey the expiration time. When the request is HTTP GET,then the ALTO server MAY use HTTP based caching mechanism. In the case of request being POST, ALTO server SHOULD use ALTO caching mechanism. To ensure consistent behaviour and reduce implementation complexity it is recommended that the mechanism defined by this memo is used for all caching scenarios.

7.2. Client behavior

ALTO cache MUST be able to identify expiration of map level or subset level information. On expiry, cache SHOULD trigger a new request to the server to refresh the expiration.

7.3. Examples

7.3.1. Network Map with Expiry Example

```
GET /networkmap HTTP/1.1
Host: alto.example.com
Accept: application/alto-networkmap+json,application/alto-error+json

HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-networkmap+json
Cache-Control: no-cache

{
  "meta" : {},
  "data" : {
    "map-vtag" : "1266506139",
    "map" : {
      "expires": "Wed, 14 Sep 2011 16:00:00 GMT",
      "PID1" : {
        "ipv4" : [
          "192.0.2.0/24",
          "198.51.100.0/25"
        ]
      },
      "PID2" : {
        "ipv4" : [
          "198.51.100.128/25"
        ]
      },
      "PID3" : {
        "ipv4" : [
          "0.0.0.0/0"
        ],
        "ipv6" : [
          "::/0"
        ]
      }
    }
  }
}
```

7.3.2. Filtered Network Map with Expiry Example

```
POST /networkmap/filtered HTTP/1.1
Host: custom.alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-networkmapfilter+json
Accept: application/alto-networkmap+json,application/alto-error+json

{
  "pids": [ "PID1", "PID2" ]
}


HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-networkmap+json

{
  "meta" : {},
  "data" : {
    "map-vtag" : "1266506139",
    "map" : {
      "PID1" : {
        "expires": "Wed, 14 Sep 2011 16:00:00 GMT",
        "ipv4" : [
          "192.0.2.0/24",
          "198.51.100.0/24"
        ]
      },
      "PID2" : {
        "expires": "Wed, 14 Dec 2011 16:00:00 GMT",
        "ipv4": [
          "198.51.100.128/24"
        ]
      }
    }
  }
}
```

8. Subscription and Notification Service

The Subcription service allows ALTO clients to subscribe to events so that ALTO server can return full maps or subset of the full maps available in the Map service. On event changes, ALTO server sends notifications for the changed events.

8.1. Requesting a Subscription

When a subscriber wishes to subscribe to a particular state for a resource, it forms a HTTP POST message.This POST message will be confirmed with a final message, 200 OK message indicating that the subscription is accepted and notification message will be sent immediately.

The framework allows entities to subscribe full or subset of full maps provided by the Map service.For example ALTO clients can subscribe for complete maps like NetworkMap, CostMaps or also for subsets like NetworkMapFilter, CostMapFilter or Endpoint services. Subscription request MUST carry "eventtype", "expires" value, "contactaddr" parameters encoded in the JSON object and sent as payload.The "eventtype" indicates what type of events subscription is required.The "expires" key indicates the actual duration for which the subscription will remain active.

Subscribers need to provide "contactaddr" in the first subscription message which will be used by server nodes to send notification messages.

Non-2xx final response indicates that no subscription is created and no notification messages will be sent.

8.1.1. Input Parameters

Input parameters are supplied in the entity body of the HTTP POST request. This document specifies the input parameters with a data format indicated by a a media type "application/ alto-subscribeinput+json", which is a JSON object of SubscribeInput:

```
object {
     JSONString event-types<0....*>;
     JSONInteger expires;
     ContactAddr [AddressType]<0..*>;
     ReqFilteredNetworkMap networkfiltermap;   [OPTIONAL]
     ReqFilteredCostMap costfiltermap;         [OPTIONAL]
     ReqEndpointProp endpointprop;             [OPTIONAL]
     ReqEndpointCostMap endpointcostmap;       [OPTIONAL]
   } SubscribeInput;
```

with members:

event-types: List of event types for which subcsription needs to be done. This can be "NetworkMap", "CostMap", "NetworkMapFilter", "CostMapFilter", "endpointprop", "endpointcost" strings. NetworkMap and CostMap events subscrobes for complete information and rest of

the event types handles subsets.

expires: The expires value indicates the lifetime of the subscription.

contactaddr: Used by the notifier to contact the subscriber for sending notification details.

All the objects are optional and needs to be added based on the eventtype parameter. Refer I-D.ietf-alto-protocol for ReqFilteredNetworkMap, ReqFilteredCostMap, ReqEndpointProp and ReqEndpointCostMap JSON object definitions.

8.2. Refreshing of Subscriptions

Subscriber can send refresh requests before a subscription expires by sending HTTP POST message with the same event and subscrid parameter returned in 200 OK message.The subscrid parameter is used to differentiate multiple subscriptions from the same node.

If a request to refresh a subscription receives a "404" response, this indicates that the subscription has already been terminated.

8.3. Notifier Subscription Handling

The Notifier should check that the event specified is understood. If not, notifier should send "503 Service Unavailable" response to indicate that the specified event is not understood.

If the event is understood, notifier creates a subscription and stores the "contactaddr", event-type and expires information.Notifier returns a "200 OK" message which indicates that the subscription is succesful. 200 OK reponse message contains the "event" information,a unique "subscrid" parameter and "expires" value. The subscrid parameter is used to differentiate multiple subscriptions.

8.3.1. Response

Input parameters are supplied in the entity body of the 200 OK message. This document specifies the input parameters with a data format indicated by a a media type "application/ alto-subscriptiondata+json".

```
object {
     JSONString subscrid;
     JSONInteger expires;
     JSONString event-types<0....*>;
   } SubcriptionData;
```

with members:

subscrid: This id is sent by notifier and is used to uniquely identify subscription. Refresh subscriptions and notifications MUST carry this id

event-types: This should be the same received in the subscription request

8.4. Notifier Notification behavior

Notifier indicates a change in the subscribed state by sending a HTTP POST message with resource information, event, "substate" and "subscrid" parameter.The resource information can be NetworkMap, CostMap, NetworkMapFilter,CostMapFilter or EndPoint Service information. The "substate" refers to the subscription state which can be "active" or "terminated". The "substate" indicates the subscription state maintained by the notifier.

If the value of the "substate" is active, the notifier should include an expires value indicating the time remaining for the subscription.

If the value of the "substate" is terminated, the notifier may include a "reason" value.

8.4.1. Input Parameters

Input parameters are supplied in the entity body of the HTTP POST request. This document specifies the input parameters with a data format indicated by a a media type "application/ alto-notificationdata+json".

```
object {
     JSONString event-type<0....*>;
     JSONString subscrid;
     JSONString substate;
     JSONInteger expires;
     InfoResourceNetworkMap networkmap;              [OPTIONAL]
     InfoResourceCostMap    costmap;                 [OPTIONAL]
     InfoResourceEndpointProperty endpointprop;      [OPTIONAL]
     InfoResourceEndpointCostMap endpointcostmap;    [OPTIONAL]
   } NotifitionData;
```

with members:

substate: Indicates the subscription state maintained by the server

expires: Indicates the time remaining for the subscription identified

by subscrid

All the objects are optional and needs to be added based on the eventtype parameter. Refer I-D.ietf-alto-protocol for InfoResourceNetworkMap, InfoResourceCostMap, InfoResourceEndpointProperty and InfoResourceEndpointCostMap JSON object definitions.

8.5. Subscriber Notification handling

Subscriber should check the subscrid returned in the notification message to check whether this is an outstanding subscription. If not, it MUST return "404 Not Found" response message.

If the substate is active, it means that the subscription is active and hence can check the expires value. Subscriber can send refresh requests if required.

8.6. Unsubscribing

Unsubscription is done by sending HTTP POST message with uri as example.com/unsubscribe and payload containing the "subscrid" which needs to be terminated.

8.7. Example

An ALTO node [P2P peer/Tracker] queries the ALTO server and contacts the peer for service. Node caches the data locally and wants to subscribe for resource state changes [Filtered Network Map].Refreshing of the subscription is done and is terminated later using unsubscribe option.In this example, the ALTO Server provides subscription, notification and unsubscribe service via a separate subdomain, "custom.alto.example.com".

```
POST /subscription HTTP/1.1
Host: custom.alto.example.com
Content-Length: [TODO]
Content-Type: alto-subscribeinput+json
Accept: application/alto-subscriptiondata+json,
        application/alto-error+json

{
  "event-types": ["NetworkMapFilter"],
  "expires": 3600,
  {
      "ipv4": [
          "addr": "192.0.2.0",
          "port": 5828
```

```
        ]
    },
    {
        "pids": [ "PID1", "PID2" ]
    }
}
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-subscriptiondata+json

{
    "subscrid": "1223subsrandstr",
    "expires" : 3600,
    "event-types": ["NetworkMapFilter"]
}
```

Change in the resource state is notified using notification message.

```
POST /notification HTTP/1.1
Host: custom.alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-notificationdata+json
Accept: application/alto-error+json

{
    "event-types": ["NetworkMapFilter"],
    "subscrid": "1223subsrandstr",
    "substate": "active",
    "expires": "3600",
    {
     "meta" : {},
     "data" : {
       "map-vtag" : "1266506139",
       "map" : {
         "PID1" : {
           "ipv4" : [
             "192.0.2.0/24",
             "198.51.100.0/25"
           ]
         },
         "PID2" : {
           "ipv4" : [
             "198.51.100.128/25"
           ]
         },
         "PID3" : {
           "ipv4" : [
```

```
              "0.0.0.0/0"
            ],
            "ipv6" : [
              "::/0"
            ]
          }
        }
      }
    }
  }

  HTTP/1.1 200 OK

  Node refreshes the subscription.

  POST /unsubscribe HTTP/1.1
  Host: alto.example.com
  Accept: application/alto-error+json
  Content-Type: application/alto-subscriptiondata+json

  {
      "subscrid": "1223subsrandstr",
      "expires" : 3600,
      "event-types": ["NetworkMapFilter"]
  }

  HTTP/1.1 200 OK

  Node terminates the subscription

  POST /unsubscribe HTTP/1.1
  Host: alto.example.com
  Accept: application/alto-error+json
  Content-Type: application/alto-subscriptiondata+json

  {
      "subscrid": "1223subsrandstr",
      "expires" : 0,
      "event-types": ["NetworkMapFilter"]
  }

  HTTP/1.1 200 OK
```

9. Detecting support for Subscription Service

An Information Resource Directory indicates to ALTO Clients which Information Resources are made available by an ALTO Server[I-D.ietf-alto-protocol]. The Information Resource Directory defined by I-D.ietf-alto-protocol will be extended to add new directory entries for Subscription and Unsubscription service.

The media-types mentions the list of all media types of information resources which will be sent by the Notifier.

9.1. Example

The following is an example Information Resource Directory returned by an ALTO Server. In this example, the ALTO Server provides subscription, notification and unsubscribe service via a separate subdomain, "custom.alto.example.com".

```
GET /directory HTTP/1.1
Host: alto.example.com
Accept: application/alto-directory+json,application/alto-error+json

HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-directory+json

{
"resources" : [
{
 "uri" : "http://alto.example.com/serverinfo",
 "media-types" : [ "application/alto-serverinfo+json" ]
}, {
 "uri" : "http://alto.example.com/networkmap",
 "media-types" : [ "application/alto-networkmap+json" ]
}, {
 "uri" : "http://alto.example.com/subscription",
 "media-types" : [
   "alto-subscriptiondata+json"
 ],
 "accepts" : [
   "application/alto-networkmapfilter+json",
   "application/alto-costmapfilter+json",
   "application/alto-endpointpropparams+json",
   "application/alto-endpointcostparams+json"
 ]
},{
 "uri" : "http://alto.example.com/notification",
 "media-types" : [
   "application/alto-networkmap+json",
   "application/alto-costmap+json",
   "application/alto-endpointprop+json",
   "application/alto-endpointcost+json"
 ],
 "accepts" : [
   "application/alto-networkmapfilter+json",
   "application/alto-costmapfilter+json",
   "application/alto-endpointpropparams+json",
   "application/alto-endpointcostparams+json"
 ]
},{
 "uri" : "http://alto.example.com/unsubscribe"
}
]
}
```

10. Transport Considerations

If subscription mechanism needs to be used the ALTO nodes MUST work as peers.

Input parameters and responses are encoded in the HTTP request's entity body, and the request uses the HTTP POST method.

11. Acknowledgements

12. IANA Considerations

12.1. application/alto-* Media Types

This document requests the registration of multiple media types, listed in Table 1.

Type	Subtype	Specification
application	alto-subscribeinput+json	Section 6.1.1
application	alto-subscriptiondata+json	Section 6.4.1
application	alto-notificationdata+json	Section 6.5.1

Table 1

13. Security Considerations

TBD

14. References

14.1. Normative References

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

14.2. Informative References

[I-D.ietf-alto-deployments] Stiemerling, M. and S. Kiesel, "ALTO Deployment Considerations", draft-ietf-alto-deployments-03 (work in progress), November 2011.

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-10 (work in progress), October 2011.

[I-D.ietf-alto-reqs]
Kiesel, S., Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", draft-ietf-alto-reqs-13 (work in progress), January 2012.

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

Authors' Addresses

Sreekanth Madhavan
Huawei Technologies
Bangalore,
India

Phone:
Email: sreekanth.madhavan@huawei.com

Nandiraju Ravishankar
Huawei Technologies
Bangalore,
India

Phone:
Email: ravi.nandiraju@huawei.com