ALTO Extension: Endpoint Cost Service for Flows

draft-wang-alto-ecs-flows-00

J. Wang\textsuperscript{1} Q. Xiang\textsuperscript{1,2}
\textsuperscript{1}Tongji University \textsuperscript{2}Yale University

October 27, 2015@ ALTO Interim Meeting
ECS: Endpoint Cost Service

- ECS request in legacy ALTO:

  ```json
  object {
    CostType cost-type;
    [JSONString constraints<0..*>;]
    EndpointFilter endpoints;
  } ReqEndpointCostMap;

  object {
    [TypedEndpointAddr srcs<0..*>;]
    [TypedEndpointAddr dsts<0..*>;]
  } EndpointFilter;
  ```
Limitation

- Use Case #1
  - ECS only allows IP addresses.
Limitation

- Use Case #2
  - Cost may vary depending on flow attributes.
Motivation

• The Definition of Flow
  – Flow: A unidirectional sequence of packets sharing
    • src-address: IP, MAC, Domain Name
    • dst-address: IP, MAC, Domain Name
    • src-vlan, dst-vlan
    • ToS
    • protocol: HTTP, FTP, SSH
    • etc ...
Motivation

• The Definition of Flow
  – Flow: A unidirectional sequence of packets sharing
    • src-address: IP, MAC, Domain Name
    • dst-address: IP, MAC, Domain Name
    • src-vlan, dst-vlan
    • ToS
    • protocol: HTTP, FTP, SSH
    • etc ...
Motivation

- The Definition of Flow
  - Flow: A unidirectional sequence of packets sharing
    - **src-address**: IP, MAC, Domain Name
    - **dst-address**: IP, MAC, Domain Name
    - **src-vlan, dst-vlan**
    - **ToS**
    - **protocol**: HTTP, FTP, SSH
    - **etc ...**

Flow Constraints
Solution

- Supporting multiple address types
  - MAC: [RFC 7024]
    - 01-23-45-67-89-AB
    - ...
  - Domain name: [RFC 1034]
    - www.alto.org
    - www.b.com
    - ...

http://snlab.org
Solution

- Including flow constraints
  - L4-protocol
    - FTP, HTTP, SSH
      - ...
  - ToS
    - Best-effort
      - ...
  - Port
    - 22, 80, 443
      - ...

object-map {
  ConstraintName -> ConstraintValue;
} FlowConstraint;
Two Design Options

- Design #1: enumerate every flow
- Design #2: combinations of address and flow constraint

Request

Design #1

```
object {
    CostType cost-type;
    [JSONString constraints<0..*>;]
    Flow flows<0..*>;
} ReqEndpointCostMap;
```

Design #2

```
object {
    CostType cost-type;
    [JSONString constraints<0..*>;]
    EndpointFilter endpoints;
} ReqEndpointCostMap;
```

```
object {
    [TypedEndpointAddr srcs<0..*>;]
    [TypedEndpointAddr dsts<0..*>;]
    [FlowConstraint flow-constraints<0..*>;]
} EndpointFilter;
```
Response

Design #1

object {  
  FlowCostMapData  
    flow-cost-map;  
  } InfoResourceEndpointCostMap :  
    ResponseEntityBase;

object-map {  
  Flow -> JSONValue;  
} FlowCostMapData;

Design #2

object {  
  EndpointCostMapData  
    endpoint-cost-map;  
  } InfoResourceEndpointCostMap :  
    ResponseEntityBase;

object-map {  
  TypedEndpointAddr ->  
    EndpointDstCosts;  
} EndpointCostMapData;

object-map {  
  TypedEndpointAddr ->  
    FlowConstraintCosts;  
} EndpointDstCosts;

object-map {  
  FlowConstraint -> JSONValue;  
} FlowConstraintCosts;
```json
{  
  "cost-type": {...},
  "flows": [
    {
      "src": "ipv4:192.0.2.2",
      "dst": "ipv4:192.1.1.2",
      "constraints": [
        {"l7-protocol": "SSH",  
         "port": "22"}]
    },
    {
      "src": "ipv4:192.0.2.2",
      "dst": "ipv4:192.1.1.2",
      "constraints": [
        {"l7-protocol": "FTP",  
         "port": "80"}]
    }
  ]
}

Design #1
```

```json
{  
  "cost-type": {...},
  "endpoints": {  
    "srcs": ["ipv4:192.0.2.2"],
    "dsts": ["ipv4:192.1.1.2"],
    "flow-constraints": [
      {"l7-protocol": "SSH",  
       "port": "22"},
      {"l7-protocol": "FTP",  
       "port": "80"}]
  }
}

Design #2
```
Example

Response

```
{
  "flow-cost-map": [
    {
      "src": "ipv4:192.0.2.2",
      "dst": "ipv4:192.1.1.2",
      "constraints": [
        {
          "l7-protocol": "SSH",
          "port": "22"
        },
        {
          "l7-protocol": "FTP",
          "port": "80"
        }
      ]
    }
  ]
}
```

```
{
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.1.1.2": {
        "l7-protocol": "SSH",
        "port": "22"}
      },
      "l7-protocol": "FTP",
      "port": "80"}
  }
}
```

Design #1

Design #2
## Compatibility Analysis

<table>
<thead>
<tr>
<th></th>
<th>Legacy ALTO Server</th>
<th>Extended ALTO Server</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Legacy ALTO Client</strong></td>
<td>Work with Design #1 and #2</td>
<td>Work with Design #1 and #2</td>
</tr>
<tr>
<td><strong>Extended ALTO Client</strong></td>
<td>Only work with Design #2</td>
<td>Work with Design #1 and #2</td>
</tr>
</tbody>
</table>
### Comparison

<table>
<thead>
<tr>
<th>Design</th>
<th>Redundancy In Request Message</th>
<th>Granularity</th>
<th>Redundancy in Response Message</th>
<th>Extra Load on ALTO Servers</th>
<th>Compatibility with Legacy ALTO Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>High</td>
<td>High</td>
<td>Medium</td>
<td>No</td>
<td>Low</td>
</tr>
<tr>
<td>#2</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Varies</td>
<td>High</td>
</tr>
</tbody>
</table>

We choose Design #2 in that it is

- consistent with design principles of ALTO;
- fully compatible with legacy ALTO.
Conclusion

- Current ECS only supports IP addresses.
- We extend ECS with flow.
  - Support multiple address types
  - Include flow constraints
- Our design uses combinations of address and flow constraint.
  - Fully compatible with legacy ALTO