

ALTO Extension: Endpoint Cost Service for Flows

`draft-wang-alto-ecs-flows-00`

J. Wang¹ Q. Xiang^{1,2}

¹Tongji University ²Yale University

October 27, 2015@ ALTO Interim Meeting

Background

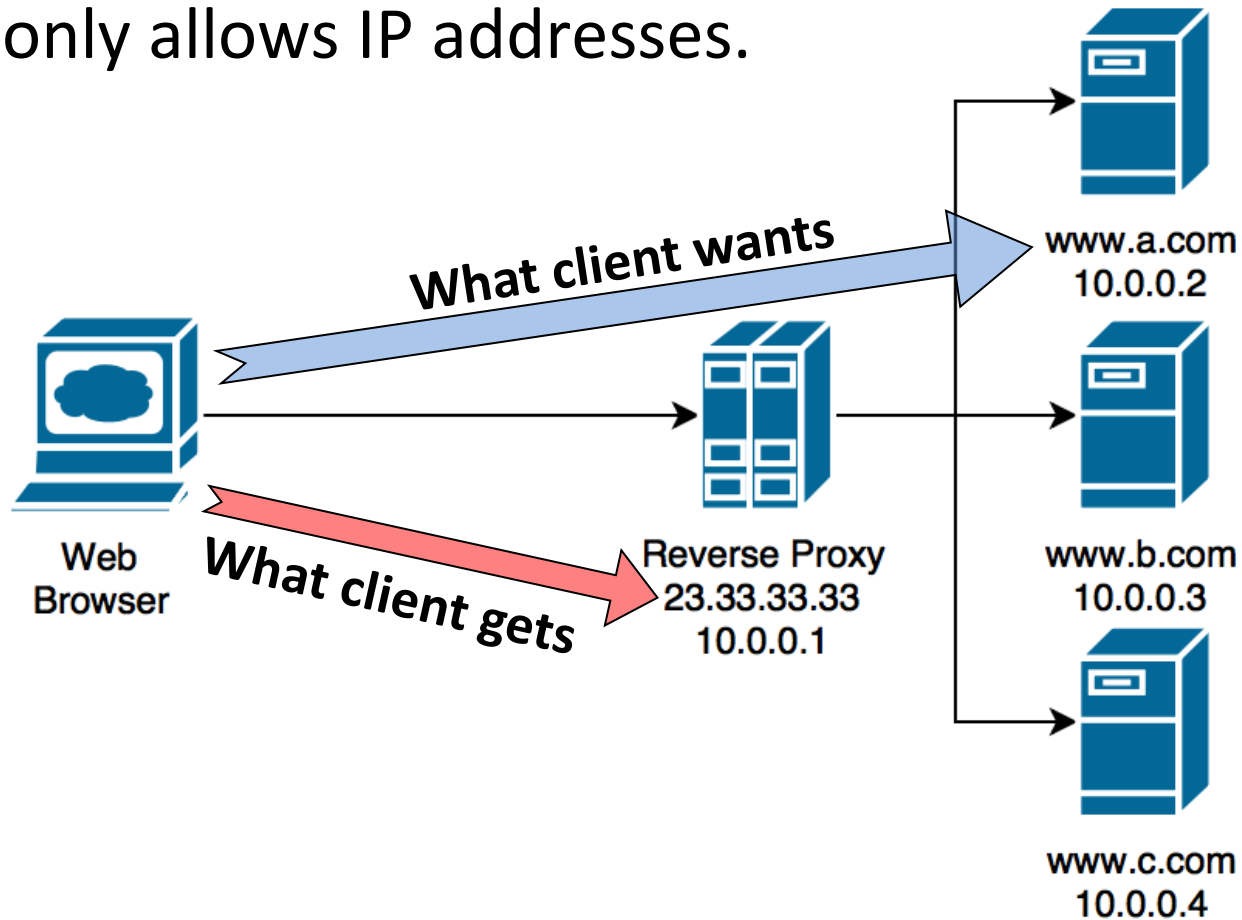
- ECS: Endpoint Cost Service
 - ECS request in legacy ALTO:

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

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

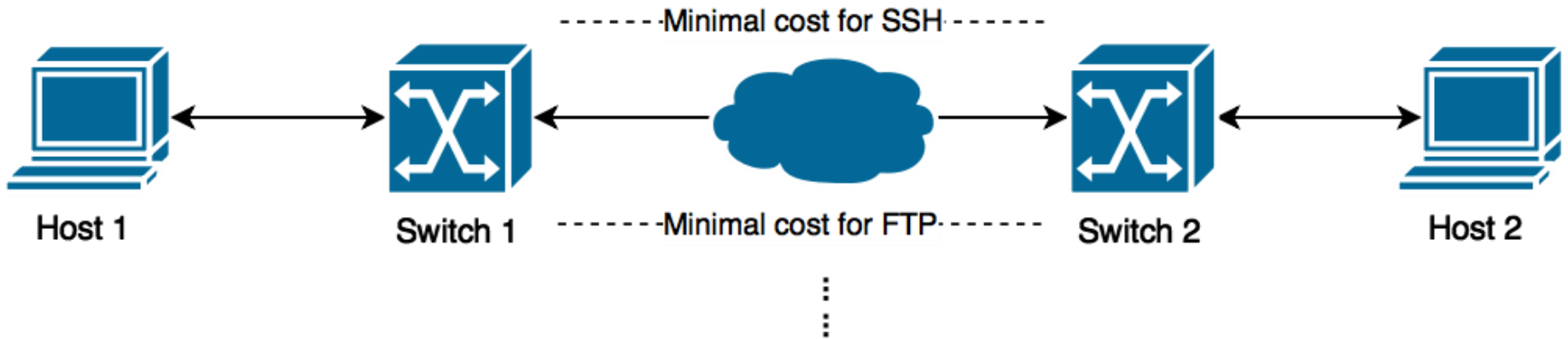
Limitation

- 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
 - ...

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

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

object {
  TypedEndpointAddr src;
  TypedEndpointAddr dst;
  FlowConstraint constraints<0..*>;
} Flow;
```

Design #1

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

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

Design #2

Two Design Options-Cont.

Response

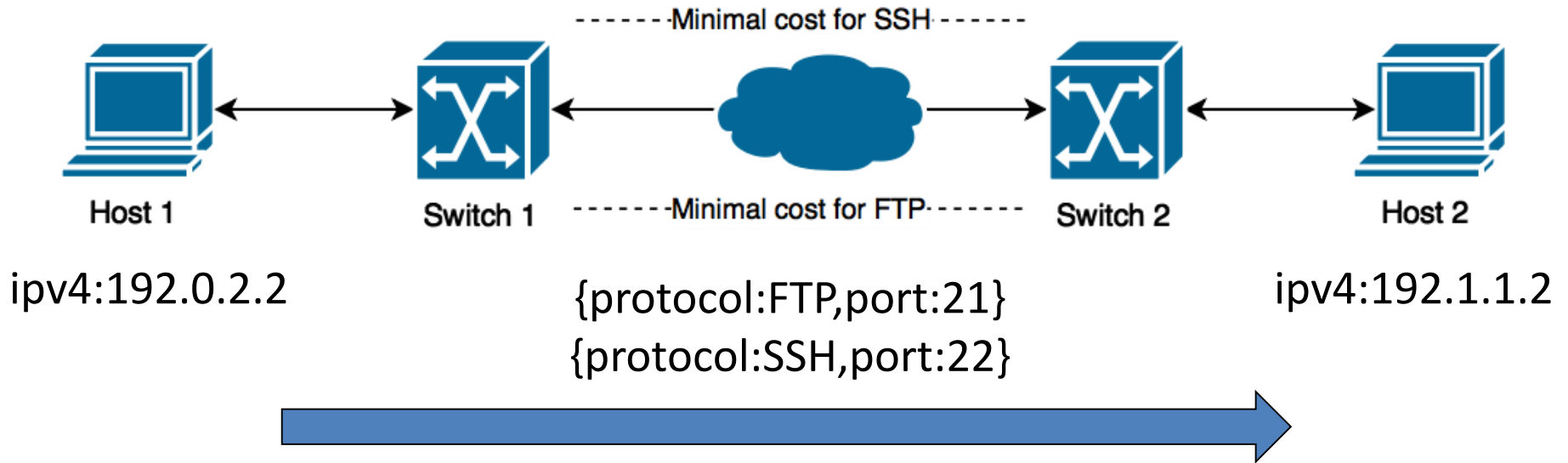
```
object {  
  FlowCostMapData  
    flow-cost-map;  
} InfoResourceEndpointCostMap :  
  ResponseEntityBase;  
  
object-map {  
  Flow -> JSONValue;  
} FlowCostMapData;
```

Design #1

```
object {  
  EndpointCostMapData  
    endpoint-cost-map;  
} InfoResourceEndpointCostMap :  
  ResponseEntityBase;  
object-map {  
  TypedEndpointAddr ->  
    EndpointDstCosts;  
} EndpointCostMapData;  
object-map {  
  TypedEndpointAddr ->  
    FlowConstraintCosts;  
} EndpointDstCosts;  
object-map {  
  FlowConstraint -> JSONValue;  
} FlowConstraintCosts;
```

Design #2

Example



Example

Request

```
{
  "cost-type": {...},
  "flows" : [
    {
      "src": "ipv4:192.0.2.2",
      "dst": "ipv4:192.1.1.2",
      "constraints" : [
        "17-protocol":"SSH",
        "port":"22"
      ],
    },
    {
      "src": "ipv4:192.0.2.2",
      "dst": "ipv4:192.1.1.2",
      "constraints" : [
        "17-protocol":"FTP",
        "port":"80"
      ]
    }
  ]
}
```

Design #1

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

Design #2

Example

Response

```
{
  "flow-cost-map": [
    {
      "src": "ipv4:192.0.2.2",
      "dst": "ipv4:192.1.1.2",
      "constraints" : [
        "17-protocol":"SSH",
        "port":"22"]
    } : 1,
    {
      "src": "ipv4:192.0.2.2",
      "dst": "ipv4:192.1.1.2",
      "constraints" : [
        "17-protocol":"FTP",
        "port":"80"]
    } : 2
  ]
}

{
  "endpoint-cost-map" : {
    "ipv4:192.0.2.2": {
      ipv4:192.1.1.2": {
        {"17-protocol" : "SSH",
        "port" : "22"} : 1,
        {"17-protocol" : "FTP",
        "port" : "80"} : 2
      }
    }
  }
}
```

Design #1

Design #2

Compatibility Analysis

	Legacy ALTO Server	Extended ALTO Server
Legacy ALTO Client	Work with Design #1 and #2	Work with Design #1 and #2
Extended ALTO Client	Only work with Design #2	Work with Design #1 and #2

Comparison

	Redundancy In Request Message	Granularity	Redundancy in Response Message	Extra Load on ALTO Servers	Compatibility with Legacy ALTO Protocol
Design #1	High	High	Medium	No	Low
Design #2	Low	Medium	Medium	Varies	High

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