Flow-based Cost Query

draft-gao-alto-fcs-03

Kai Gao[1] J. Jensen Zhang[2]
Qiao Xiang[3] Y. Richard Yang[3]

[1] Tsinghua University [2] Tongji University [3] Yale University

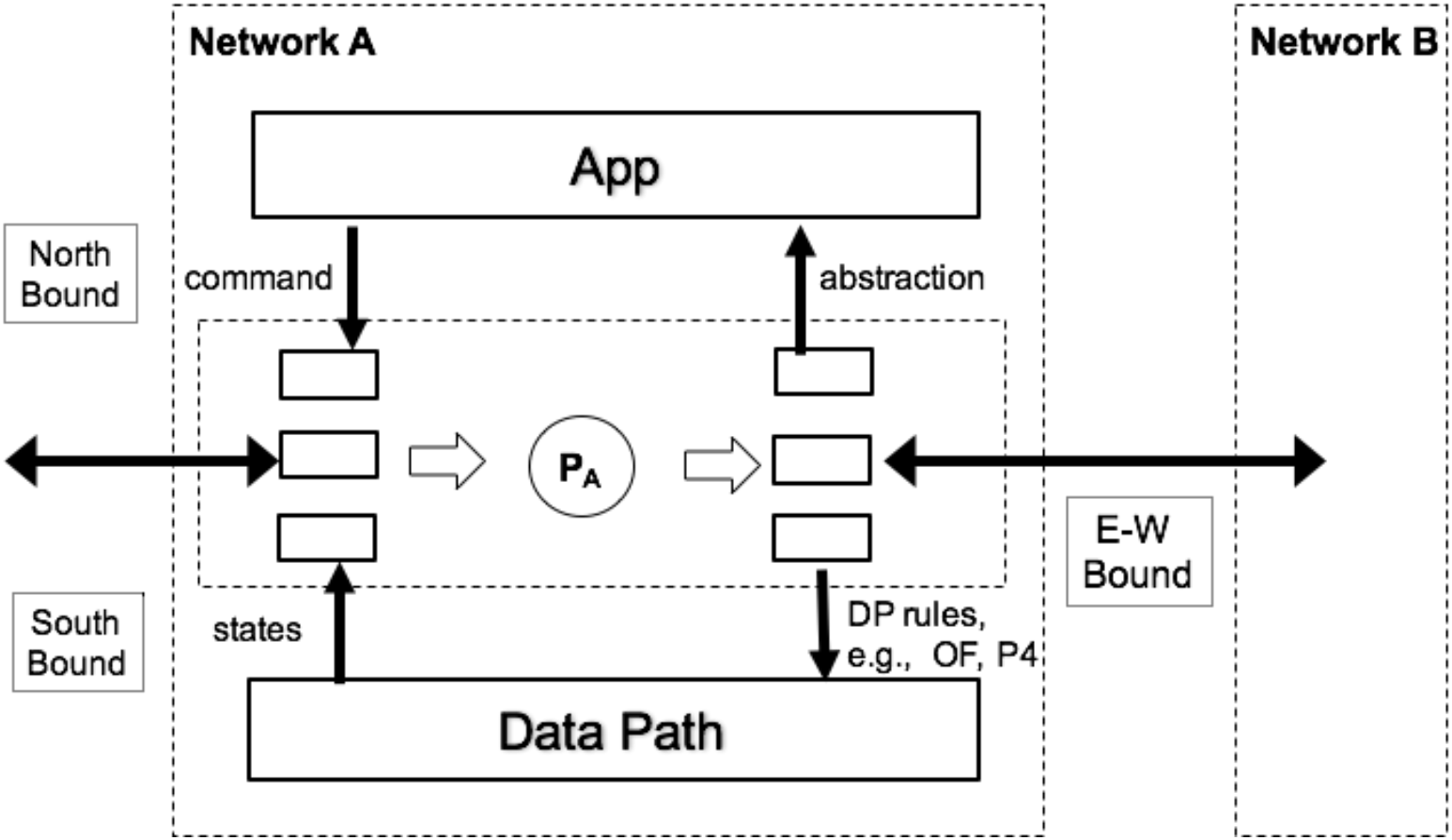
Presenter: Jensen

IETF 99, July 20, 2017, Prague

Updates: Overview

- Many updates from -01 (March 13, 2017, IETF 98) to -03 (July 03, 2017, IETF 99)
  - Added an architecture section, focusing on using ALTO as a base to distribute network information resources for SDN networks
  - Many syntax revisions to make the design more extensible, e.g.,
    - Change the schema of "pid-flows" and "endpoint-flows" fields from pair list to pair mesh list.
    - Change "EndpointURI" to "AddressType::EndpointAddr" for consistency.
    - Replace "Cost Confidence" by "Cost Statistics" for compatibility.

Architecture: ALTO Providing Unified NorthBound/East-West Views



Big Picture: Unified Model-Views in SDN

ALTO Function: **Network information space → View**

Model-views mapping of different ALTO query services:

- Filtered Network Map Service:
  1-dimensional group region → endpoint set
- Filtered Endpoint Property Service:
  1-dimensional address region → property view
- Filtered Cost Map Service:
  2-dimensional rectangular group region → cost view
- Endpoint Cost Service:
  2-dimensional rectangular address region → cost view

Requirements of Flow-based Query

General Requirements of the Unified Interface:

- **More flexible input**: Target of FCS
- **More flexible output**: Target of Path Vector, Unified Property, Multi-Cost, Cost Calendar

FCS Requirements:

- **#1** More flexible shape of query space
- **#2** More expressive encoding of query entry

Design Decisions

- #1 Query schema: addr-based vs. spec-based
- #2 Entry encoding: type:addr vs. header-field
- #3 Validation: error or inheritance

Current decisions:

- Co-existence:
  - addr-based + extended type:addr for legacy media-type
  - spec-based + header-field for new media-type
- Return ERROR for all invalid queries

Trade-off between addr-based and spec-based

Extended Legacy Cost Query Schema (address-based schema):

```
object {
  [CostType cost-type;]
  [CostType multi-cost-types<1..*>;]
  [CostType testable-cost-types<1..*>;]
  [JSONString constraints<0..*>;]
  [JSONString or-
   constraints<1..*><1..*>;]
} MultiCostRequestBase;

object {
  [EndpointFilter endpoints;]
  [EndpointFilter endpoint-flows<1..*>;]
} ReqEndpointCostMap :
MultiCostRequestBase;
```

FCS Query Schema (specification-based schema):

```
object {
  FlowFilterMap      flows;
} FlowCostRequest :
MultiCostRequestBase;

object-map {
  FlowId -> FlowFilter;
} FlowFilterMap;
```

Trade-off between type:addr and header-field

Compatible Query Entry Descriptor:
AddressType:EndpointAddr

New ALTO Address Type Registry (Section 8.1 of draft-gao-alto-fcs-03)

Valid query entries:

```
"eth:98-e0-d9-9c-df-81"
"http:www.example.com"
"ftp:198.51.100.34:5123"
"tcp:[2000::1:2345:6789:abcd]:8080"
```

Address type conflict:

```
{
  "srcs": ["ftp:192.168.0.2:5123"],
  "dsts": ["http:www.google.com"]
}
```

New Query Entry Descriptor:

```
object-map {
  TypedHeaderField -> JSONValue;
} FlowFilter;
```

Valid query entry:
(We can define a query entry without any information about the source point.)

```
{
  "ipv4:dst": "192.168.1.3",
  "tcp:dst": 22,
  "eth:vlan-id": 20
}
```

Remaining Issue: Fault Tolerance

Consider the following query:

```
"endpoint-flows":  [
  {
    "srcs":  ["ipv4:192.0.2.2"],
    "dsts":  ["ipv4:192.0.2.89",
              "http:cdn1.example.com"]
  }, ... (1)
  {
    "srcs":  ["udp:203.0.113.45:54321"],
    "dsts":  ["http:cdn1.example.com"]
  }  ... (2)
]
```

Only filter **(2)** conflicts, but the ALTO server won't return the cost of **(1)**.

The ALTO client has to re-send **(1)** in the revised query.

Is it possible to return the response of the valid part with the error message for the invalid part?

Option 1: Augment error message into the [endpiont]cost-map response.

Option 2: Automatic conflict avoidance.

e.g. `"udp"` is a specific type of `"ipv4"/"ipv6"`, so the ALTO server reduce the src endpoint address to `"ipv4:203.0.113.45"` and return the cost between it and `"http:cdn1.example.com"`.

Next Steps

- Finalize design decisions
- Clarify use cases

Backup Slides