

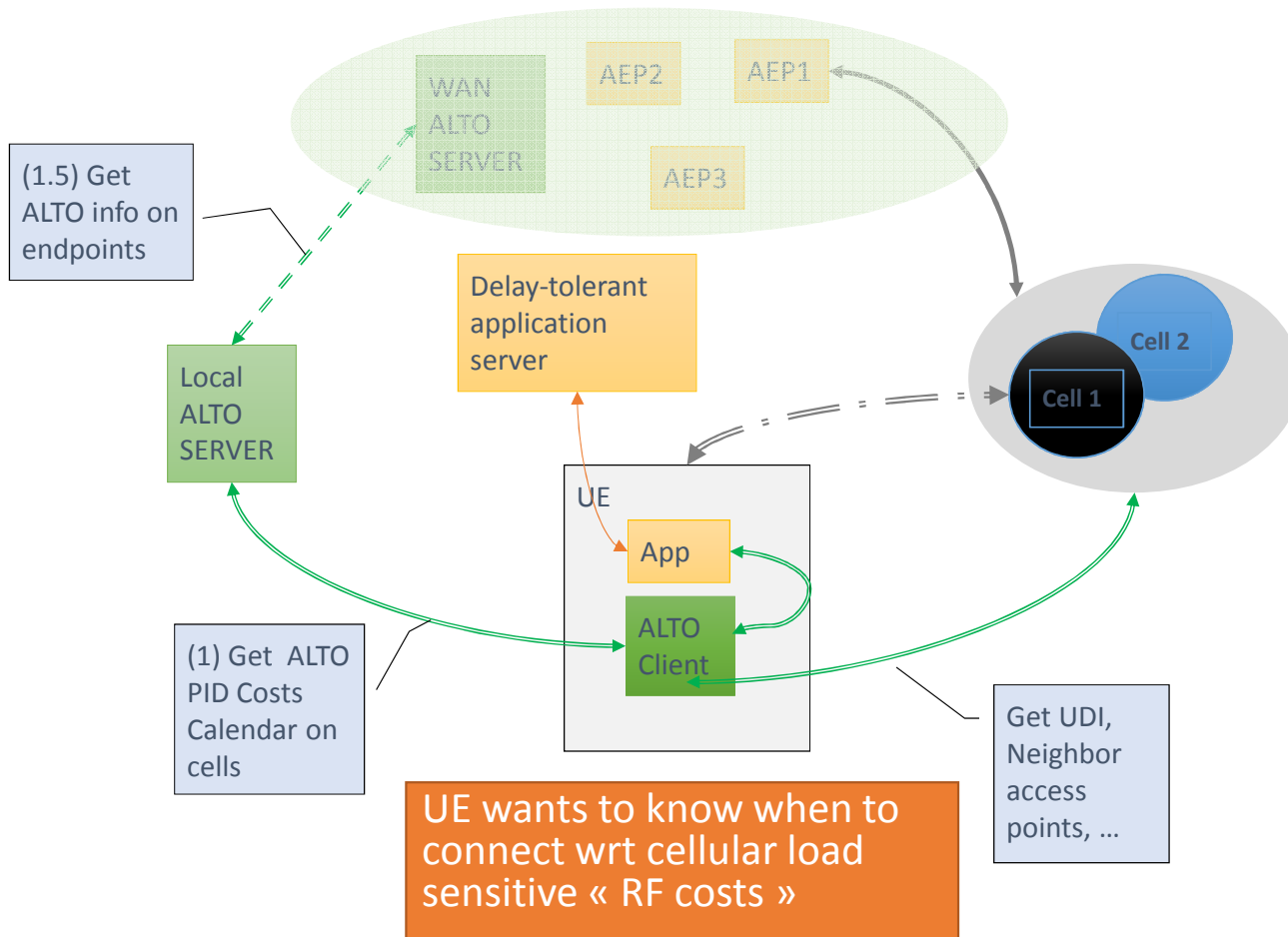
# draft-randriamasy-alto-cost-context-01

## Protocol extension proposal

ALTO WG meeting IETF98 March 31, 2017

S. Randriamasy

# Use case 1



UDI= Unattended Data Indicator

« Simple » ALTO Cost Calendar

5	1	3	2	5
---	---	---	---	---

Other network information eg LTE release 13 may **instantly** indicate whether unattended data transmission is allowed or not.  
 → UE may do finer grained transmission/connection decisions

# Example: context\_param = [uda, udna]

## « Simple » ALTO Calendar

5	1	3	2	5
---	---	---	---	---

## « UDI aware » ALTO Calendar

4	1	2	1	4	UDNA
5	2	4	3	5	UDA

- UDA = Unattended Data Allowed
  - Indicates good connection
- UDNA = Unattended Data Not Allowed
  - Indicates poor connection

**ALTO Calendar values = NON REAL TIME**  
**UDI = REAL TIME provided by network**

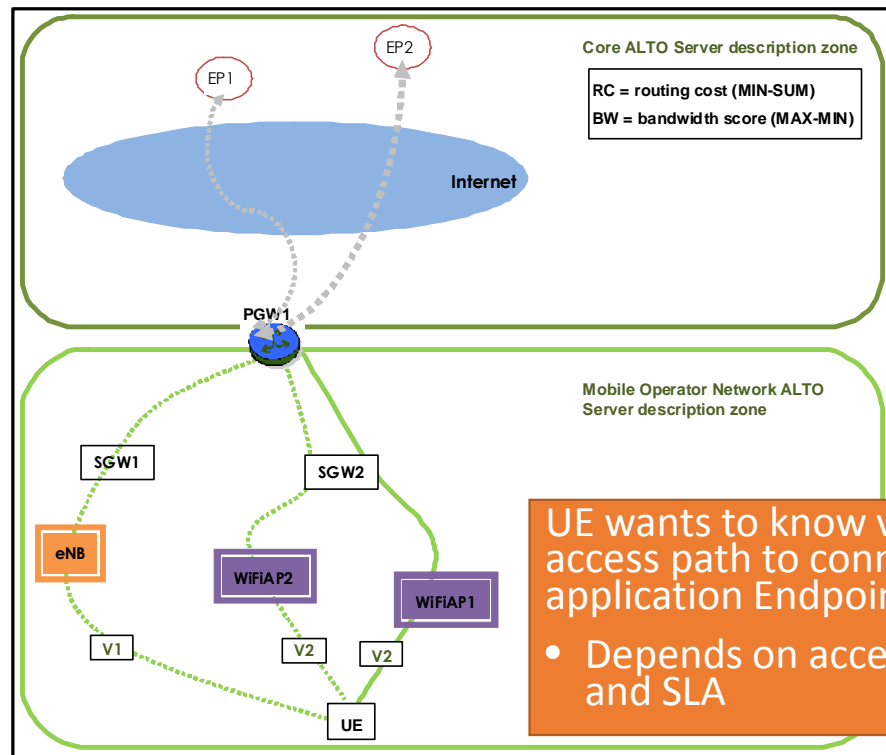
## EXAMPLE

- UDI sent every 1 msec:
  - = UDA when RFCost  $\geq$  3
  - = UDNA otherwise
- ALTO Calendar attributes
  - Nb time intervals: 5
  - Time interval length: 1-5 minutes

## PROBLEM

- **Cumbersome to define a metric associated to context C1, C2, ... CN**
- **Request input may be too heavy**
- **Not affordable to get N cost maps**

## Use case 2



UE wants to know via which access path to connect to an application Endpoint

- Depends on access type and SLA

## Use case 3

ALTO Client wants Cost Maps with values depending on different policies

- Discussion launched in 2012 for CDN applications
- Cost Maps for CDN can be voluminous

**PROBLEM** for cases 2 and 3

- Cumbersome to define a metric associated to context C1, C2, ... CN
- Request input may be too heavy
- Not affordable to get N cost maps

# Proposal

- **Extend cost information specified in [RFC7285] by**
  - providing, *for a same cost metric, several possible cost values*
  - where each value depends on *qualitative* criteria as opposed to quantitative criteria such as time.
- **Approach: introduce member “cost-context”**
  - in IRD resource capabilities + “meta” of ALTO queries and responses
  - “cost-context” has members
    - “cost-type-names” in IRD
    - “context-params” = [[P1, P2, ... ] AND [CA, CB, ...].... ], list of possible parameter combinations
- **Thus allowing for example**
  - **Ex 1:** get an ALTO calendar of cellular connection costs
    - In the uplink and downlink directions AND
    - Depending on whether unattended data is allowed or not.
  - **Ex 2:** getting "routingcost" values
    - In the case of cellular and wifi access type AND
    - In case of “SLA3”
  - **Ex 3:** get Cost Maps with values for policy, Pa, Ph, Pw, ...

## Applicable ALTO Services:

- **Filtered Cost Map**
- **Endpoint Cost Service**

# Example IRD

```
"filtered-cost-calendar-map" : {  
  "uri" : "http://alto.local.example.com/costmap/filtered/calendar/context",  
  "media-types" : [ "application/alto-endpointcost+json" ],  
  "accepts" : [ "application/alto-endpointcostparams+json" ],  
  "capabilities" : {  
    "cost-constraints" : true,  
    "cost-type-names" : [ "num-routingcost", "num-RFcost" ],  
    "calendar-attributes" : [  
      {"cost-type-names" : "num-routingcost", "time-interval-size" : "1 hour", "number-of-intervals" : 24},  
      {"cost-type-names" : "num-RFcost", "time-interval-size" : "5 minute", "number-of-intervals" : 12}  
    ],  
    "cost-context" : [ // ++NEW  
      {"cost-type-names" : "num-RFcost",  
        "context-params" : [{"uda", "udna "}, {"uplink", "downlink"}] // multiple occurrences ignored  
      }  
    ]  
  }  
}
```

ALTO Client may pick e.g.  
*(uda AND uplink) OR (udna AND downlink)*

AND-combination

OR-combination

# Example use case 1: request and response

```
POST /costmap/filtered/calendar/context HTTP/1.1
Host: alto.example.com
Accept: application/alto-costmap+json,application/alto-error+json
Content-Type: application/alto-costmapfilter+json
Content-Length: ###
{
  "cost-type" : { "cost-mode": "numerical", "cost-metric": "RFcost"},
  "calendared" : true,
  "context-params" : [{"uda", "uplink"},
                      ["uda", "downlink"],
                      ["udna", "uplink"],
                      ["udna", "downlink"]],
  "pids" : [
    {"srcs" : ["Cell1"], "dsts" : ["Cell1", "Cell2"]},
    {"srcs" : ["Cell2"], "dsts" : ["Cell1", "Cell2"]}
  ]
}
```

All context-param combinations selected

```
HTTP/1.1 200 OK
Content-Type: application/alto-costmap+json
Content-Length: ###
{
  "meta" : {
    "dependent-vtags" : [
      {"resource-id": "my-default-network-map",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"}
    ],
    "cost-type" : {"cost-mode": "numerical", "cost-metric": "RFcost"},
    "calendar-response-attributes" :
      { "calendar-start-time" : Tue, 1 Sept 2016 13:00:00 GMT,
        "time-interval-size" : "5 minute",
        "numb-intervals" : 12},
    "context-params" : [{"uda", "uplink"}, // ++NEW
                       ["uda", "downlink"],
                       ["udna", "uplink"],
                       ["udna", "downlink"]}
  } // end meta
  "cost-map" : {
    "Cell1": { "Cell1": [[70, 20, 90, 20], ... ,[50, 20, 70, 20]],
              "Cell2": { "Cell2": [[20, 70, 20, 90], ... ,[20, 50, 20, 70]]
            }
  }
}
```

## Example use case2: IRD and response

```
"resources" : {
  "filtered-cost-calendar-map" : {
    "uri" : "http://###/endpointcostmap/lookup/context",
    "media-types" : [ "application/alto-endpointcost+json" ],
    "accepts" : [ "application/alto-endpointcostparams+json" ],
    "capabilities" : {
      "cost-constraints" : true,
      "cost-type-names" : [ "num-routingcost",
        "num-bandwidthscore"],
      "cost-context" : [// ++NEW
        {"cost-type-names" : "num-routingcost",
          "context-params" : [ ["cell", "wifi", "lan"],
            ["SLA-1", "SLA-2", "SLA-3"]}
        ]
      }
    }
  }
} // end ECM capab
... other resources ...
} // end resources
```

If Client requests 2 combos:

- « cell » AND « SLA-3 »,
- « wifi » AND « SLA-3 »

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-endpointcost+json
{
  "meta" : {
    "cost-type" : {"cost-mode" : "numerical",
      "cost-metric" : "routingcost"},
    "context-params" : [ ["cell", "wifi"], ["SLA-3"] ]
  }
  "endpoint-cost-map" : {
    "ipv4:192.0.2.2" : {
      "ipv4:192.0.2.89" : [10, 4],
      "ipv6:2000::1:2345:6789:abcd" : [4, 6]
    }
  }
}
```

Array of 2 context-based values:  
[cell AND sla, wifi AND sla]



# Required ALTO extensions for use cases in draft

- Cost value context parameters: a capability to allow exposing several possible context-dependent values for one metric,
  - **Focus of this draft**
- Extended input for the Filtered Cost Map Service: to allow the input to comprise several(source-array, destination-array) pairs,
  - has been proposed in [draft-yang-alto-path-vector]
- Cost metrics for cellular and wireless networks: these features would extend current proposals in the WG,
  - could be added to [draft-ietf-alto-performance-metrics]
- For cellular and wireless networks: entities with associated address space and properties
  - could be added to [draft-roome-alto-unified-props]

# Design principles & choices

- Keep backwards compatibility wrt RFC7285
- No new media type
- Logically combines several sets of context parameters, moderating thus the set of conveyed parameters combinations and metrics
- Design option for cellular networks is to map a cell to a PID
  - Connection costs within a cell C1 mapped to say PID1 are conveyed as a PID1 to PID1 cost

# Next steps

- Correct IRD example
- Rules to arrange logical combinations of context params
- Examples for the EP property service
- Collect WG feedback on this proposal
  - Adopt extension supporting Cost/Property Context Parameters ?
  - Opinion on the design ?
  - Suggestions ?

# Thank you

Back-up follows

# ALTO Cost Context overview

- Resumes previous discussions and proposals
  - Discussion in 2012 on how to get Cost Map K associated to policy Y
- Applicable to many network types
- Use cases in this draft: in cellular and wireless networks.
  - assumes the availability of cellular cost metrics and associated namespace.
- Allows finer grained decisions
- To avoid defining as many metrics as context parameters
  - Instead of: cost\_policyA, ... cost\_policyT, CCost\_uda, Ccost\_udna, ...
  - → cost\_policy[pA, ... pT], Ccost[uda, udna]
- Avoid heavy request input, e.g.
- Applicable ALTO services
  - Filtered Cost Map service, Endpoint Property Service