Predicted Units & Policy Groups

IETF 99
July 2017
(draft-bertz-dime-predictunits-02 &
draft-bertz-dime-policygroups-04)

Predicted Units v02 - Motivation

- In virtual world (NFV) resources allocated to a Diameter Client may not be consistent or visible to the Client
- During a Service Authorization, e.g. RFC 4006 CCA success, letting the Client know how much resources may be consumed & by when (if that is known) would
  - Give the Client an idea of when it may become 'exhausted'
  - Give the Client an opportunity to signal to other systems its load / availability or resize itself to meet demands
    - Resizing is never automated (it takes some seconds)

Predicted Units v02 - Update

- All editorial
  - Section Header Corrections
  - More discussion on Clients

- Refresher – 2 Grouped AVPs commonly sent when Authorization of a Service occurs
  - Predicted-Service-Units ::= < AVP Header: TBD1 >
  - [CC-Time]
  - [CC-Money]
  - [CC-Total-Octets]
  - [CC-Input-Octets]
  - [CC-Output-Octets]
  - [CC-Service-Specific-Units]
  - [Time-Of-Day-Condition] < Conditions
  - *[AVP]
  -
  - Predicted-Service-Units-Series ::= < AVP Header: TBD2 > (List of PSUs with various Units / TOD conditions)
  - 1*{ Predicted-Service-Units }
- Ask – Can we start WG adoption discussion (on list)

Intermission

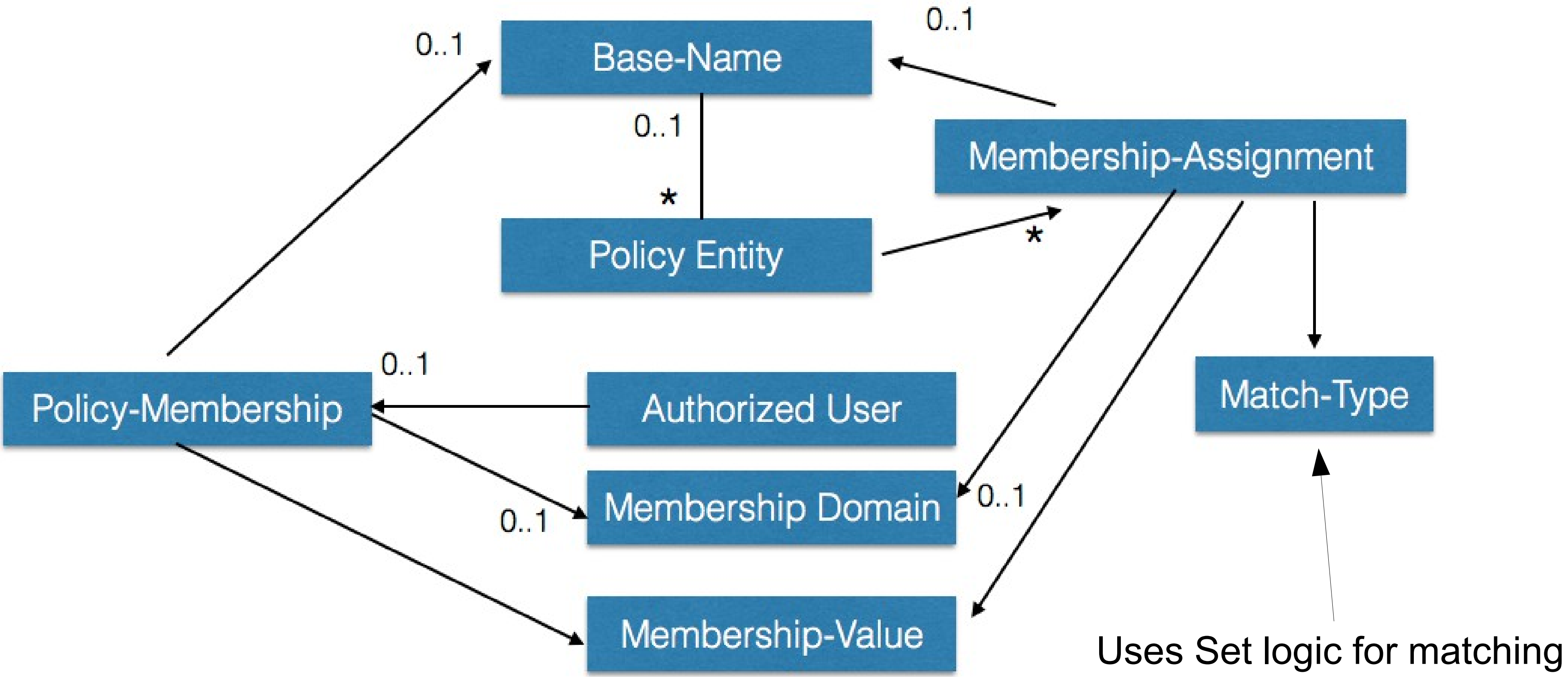
Purpose

- Provide ability to Group Policy Rules into a common name
  - Already have some Identifiers
    - Filter-Ids (UTF-8 Strings) in RFC 7155,
      - It is a list of Filter-Identifiers
      - if RADIUS support is NOT required IPFilterRule is recommended
    - Classifier-ID (OctetString) in RFC 5777
    - No consistent Rule-Id support (some in 3GPP but not all Rule Types get Rule-Ids)
  - Grouping is supported by hierarchy in 3GPP through use of Base-Name < Adopted in this proposal
- Provide an efficient mechanism for applying groups of Policy Rules that appear in multiple hierarchies
  - Akin to Charging-Characteristics used in 3GPP
  - Generalized
  - Meant to be leveraged for provisioning patterns

Policy Groups - Update

- Editorial Update
  - Reworked Examples
  - Reworked Introduction per feedback from IETF 98
    - It was large update
    - More direct discussion

Relationship Model



Policy Groups Example 1, Overlap Deduplication at Enforcement Point - Adding Membership Assignment to Filters

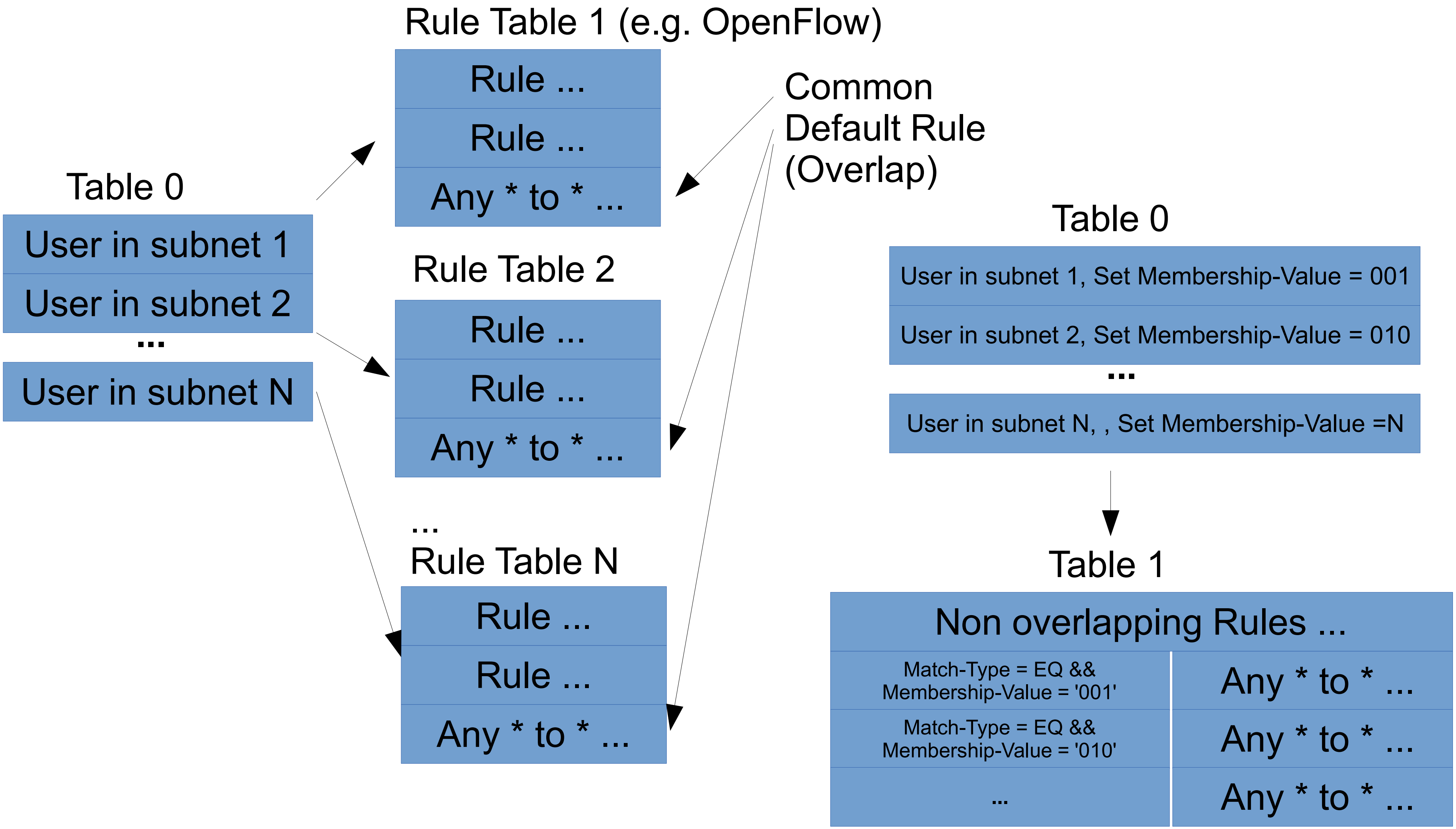


Table 0

User in subnet 1

User in subnet 2

...

User in subnet N

Rule Table 1 (e.g. OpenFlow)

Rule ...

Rule ...

Any * to * ...

Rule Table 2

Rule ...

Rule ...

Any * to * ...

...

Rule Table N

Rule ...

Rule ...

Any * to * ...

Common Default Rule (Overlap)

Table 0

User in subnet 1, Set Membership-Value = 001

User in subnet 2, Set Membership-Value = 010

...

User in subnet N, , Set Membership-Value =N

Table 1

Non overlapping Rules ...

Match-Type = EQ &&
Membership-Value = '001'
	Any * to * ...

Match-Type = EQ &&
Membership-Value = '010'
	Any * to * ...

...	Any * to * ...

Policy Groups Example 2, Application at the Decision Point Process

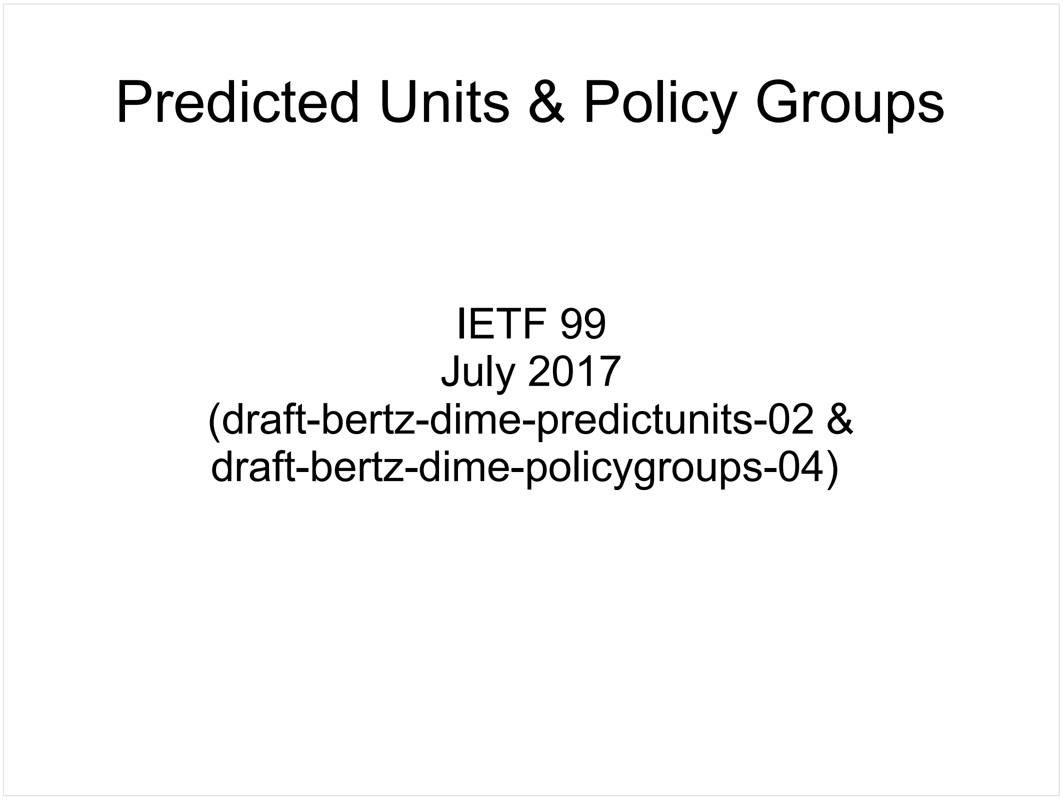
- Step 1 – Determine All Rules where Membership-Assignment (Policy) and Policy-Membership (User) match
- Step 2 – Filter Rules by Time of Day (RFC 5777) to determine active rules
- Step 3 – Communicate Active Rules to Enforcement Point

Applications

- De-duplication of Rules at the enforcement point
- Determining applicable Rules at the Decision point
- Combining both techniques (not discussed here)
- Real world scenario @ Operator
  - 60M devices with ~ 1200 rules (or so we thought)
  - We had no idea how to do this on Openflow (we were limited to 13 tables for everything...)
  - Used
    - METADATA for Membership-Value
    - Binary mask for METADATA based upon the Match-Type
  - Result
    - 3 Table Design for mobility downlink
      - Table 1 – Match by DST IP, set Metadata for packet
      - Table 2 – Match Flows and Metadata. Assign to pseudo tunnel IDs
      - Table 3 – Match by DST & pseudo tunnel ID, Apply meters, accounting and drop / tunnel / etc as required
    - 2 Table Design for mobility uplink
      - Table 1 – Unpack from Tunnel. Apply meters & accounting. Assign Metadata for packet.
      - Table 2 – Match metadata & packet header. Apply actions as required.
    - Those 1200 rules became ~ 200. More work reduced this to ~ 12 rules in Table 1 of Downlink and Table 2 of uplink.

Next Steps

- Want more reviews
- WG Adoption?



Predicted Units & Policy Groups

IETF 99
July 2017
(draft-bertz-dime-predictunits-02 &
draft-bertz-dime-policygroups-04)

Predicted Units v02 - Motivation

- In virtual world (NFV) resources allocated to a Diameter Client may not be consistent or visible to the Client
- During a Service Authorization, e.g. RFC 4006 CCA success, letting the Client know how much resources may be consumed & by when (if that is known) would
  - Give the Client an idea of when it may become 'exhausted'
  - Give the Client an opportunity to signal to other systems its load / availability or resize itself to meet demands
    - Resizing is never automated (it takes some seconds)

Predicted Units v02 - Update

- All editorial
  - Section Header Corrections
  - More discussion on Clients

- Refresher – 2 Grouped AVPs commonly sent when Authorization of a Service occurs
  - Predicted-Service-Units ::= < AVP Header: TBD1 >
  - [CC-Time]
  - [CC-Money]
  - [CC-Total-Octets]
  - [CC-Input-Octets]
  - [CC-Output-Octets]
  - [CC-Service-Specific-Units]
  - [Time-Of-Day-Condition] < Conditions
  - *[AVP]
  -
  - Predicted-Service-Units-Series ::= < AVP Header: TBD2 > (List of PSUs with various Units / TOD conditions)
  - 1*{ Predicted-Service-Units }
- Ask – Can we start WG adoption discussion (on list)

Intermission

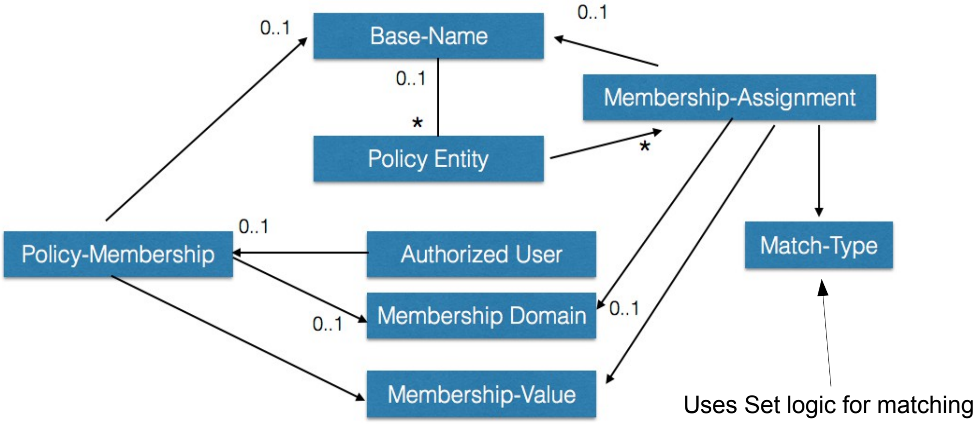
Purpose

- Provide ability to Group Policy Rules into a common name
  - Already have some Identifiers
    - Filter-Ids (UTF-8 Strings) in RFC 7155,
      - It is a list of Filter-Identifiers
      - if RADIUS support is NOT required IPFilterRule is recommended
    - Classifier-ID (OctetString) in RFC 5777
    - No consistent Rule-Id support (some in 3GPP but not all Rule Types get Rule-Ids)
  - Grouping is supported by hierarchy in 3GPP through use of Base-Name < Adopted in this proposal
- Provide an efficient mechanism for applying groups of Policy Rules that appear in multiple hierarchies
  - Akin to Charging-Characteristics used in 3GPP
  - Generalized
  - Meant to be leveraged for provisioning patterns

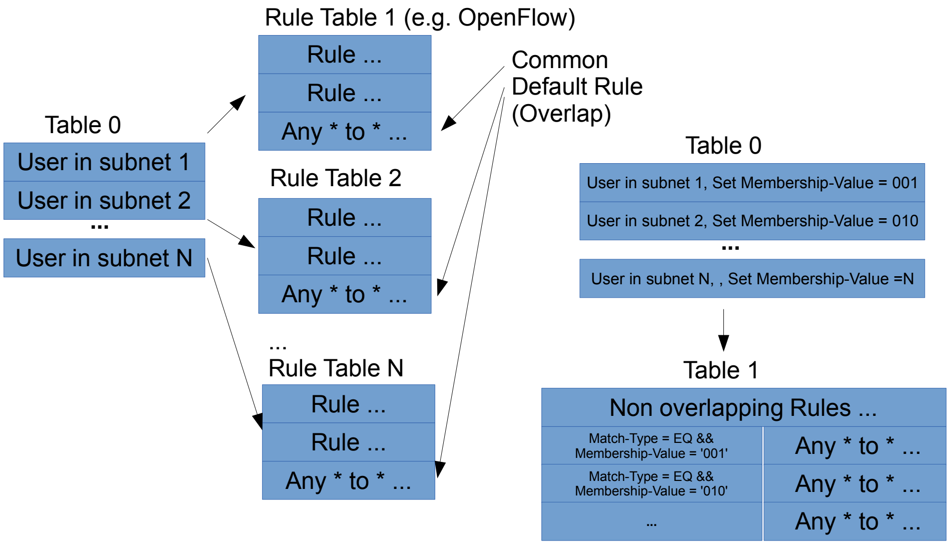
Policy Groups - Update

- Editorial Update
  - Reworked Examples
  - Reworked Introduction per feedback from IETF 98
    - It was large update
    - More direct discussion

Relationship Model



Policy Groups Example 1, Overlap Deduplication at Enforcement Point - Adding Membership Assignment to Filters



Rule Table 1 (e.g. OpenFlow)

Rule ...
Rule ...
Any * to * ...

Common
Default Rule
(Overlap)

Table 0

User in subnet 1
User in subnet 2
...
User in subnet N

Rule Table 2

Rule ...
Rule ...
Any * to * ...

...
Rule Table N

Rule ...
Rule ...
Any * to * ...

Table 0

User in subnet 1, Set Membership-Value = 001
User in subnet 2, Set Membership-Value = 010
...
User in subnet N, , Set Membership-Value =N

Table 1

Non overlapping Rules ...	

Match-Type = EQ &&
Membership-Value = '001'
	Any * to * ...

Match-Type = EQ &&
Membership-Value = '010'
	Any * to * ...

...	Any * to * ...

Policy Groups Example 2, Application at the Decision Point Process

- Step 1 – Determine All Rules where Membership-Assignment (Policy) and Policy-Membership (User) match
- Step 2 – Filter Rules by Time of Day (RFC 5777) to determine active rules
- Step 3 – Communicate Active Rules to Enforcement Point

Applications

- De-duplication of Rules at the enforcement point
- Determining applicable Rules at the Decision point
- Combining both techniques (not discussed here)
- Real world scenario @ Operator
  - 60M devices with ~ 1200 rules (or so we thought)
  - We had no idea how to do this on Openflow (we were limited to 13 tables for everything...)
  - Used
    - METADATA for Membership-Value
    - Binary mask for METADATA based upon the Match-Type
  - Result
    - 3 Table Design for mobility downlink
      - Table 1 – Match by DST IP, set Metadata for packet
      - Table 2 – Match Flows and Metadata. Assign to pseudo tunnel IDs
      - Table 3 – Match by DST & pseudo tunnel ID, Apply meters, accounting and drop / tunnel / etc as required
    - 2 Table Design for mobility uplink
      - Table 1 – Unpack from Tunnel. Apply meters & accounting. Assign Metadata for packet.
      - Table 2 – Match metadata & packet header. Apply actions as required.
    - Those 1200 rules became ~ 200. More work reduced this to ~ 12 rules in Table 1 of Downlink and Table 2 of uplink.

Next Steps

- Want more reviews
- WG Adoption?