Notification Capabilities Model Extension for Data Node Tag Capability Support

draft-tao-netconf-notification-node-tag-capabilities-00

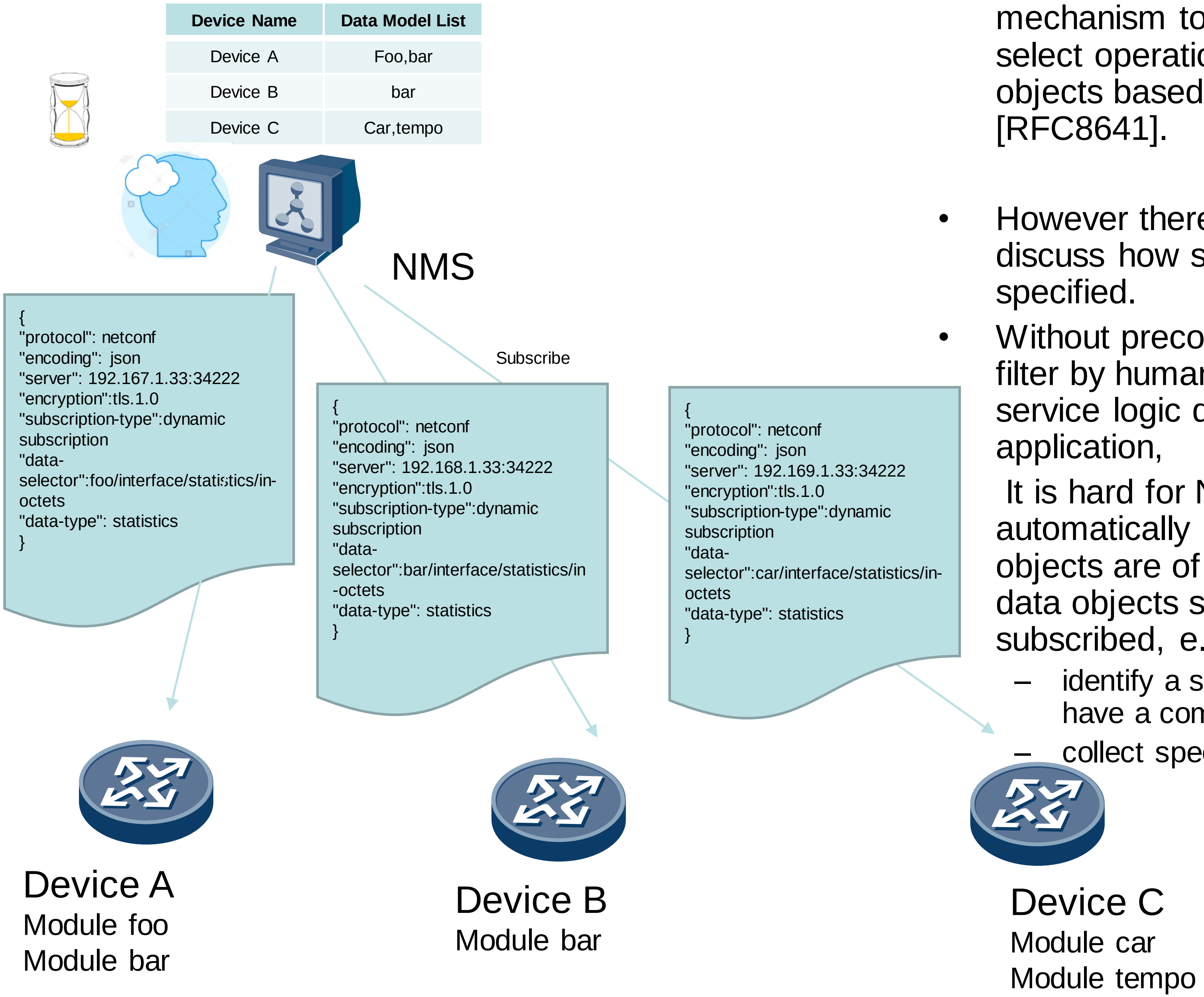
Ran Tao

Bo Wu

Background

- Notification capability model allows a client to discover YANG-Push related server capabilities such as;
  - Maximum number of objects that can be sent in an update
  - Supported (reporting) periods for periodic subscriptions
- It is defined in [I-D.netconf-notification-capabilities]
- Without using notification capability, it leads to unexpected failure or additional message exchange for NETCONF clients to discover data models supported by a NETCONF server.
- When the state of all subscriptions of a particular Subscriber to be fetched is huge, filtering queries of operational state on a server based on server capabilities can greatly reduce the amount of data to be streamed out to the destination.

Why this draft



Device Name	Data Model List
Device A	Foo,bar
Device B	bar
Device C	Car,tempo

NMS

Subscribe

```
{
"protocol": netconf
"encoding": json
"server": 192.167.1.33:34222
"encryption":tls.1.0
"subscription-type":dynamic subscription
"data-selector":foo/interface/statistics/in-octets
"data-type": statistics
}
```

```
{
"protocol": netconf
"encoding": json
"server": 192.168.1.33:34222
"encryption":tls.1.0
"subscription-type":dynamic subscription
"data-selector":bar/interface/statistics/in-octets
"data-type": statistics
}
```

```
{
"protocol": netconf
"encoding": json
"server": 192.169.1.33:34222
"encryption":tls.1.0
"subscription-type":dynamic subscription
"data-selector":car/interface/statistics/in-octets
"data-type": statistics
}
```

Device A
Module foo
Module bar

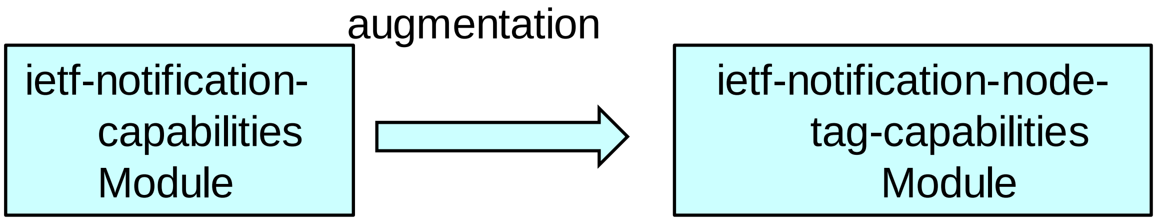
Device B
Module bar

Device C
Module car
Module tempo

- YANG push telemetry provide a mechanism to subscribe to and select operational state data objects based on selection filter [RFC8641].
- However there is no document to discuss how selection filter is specified.
- Without preconfigured selection filter by human intervening or service logic defined by the application,

  It is hard for NETCONF clients to automatically select which data objects are of interest or which data objects should be subscribed, e.g.,
  - identify a set of objects which have a common characteristic,
  - collect specific object type nodes.

Data Node Tag Capability Proposal



augmentation

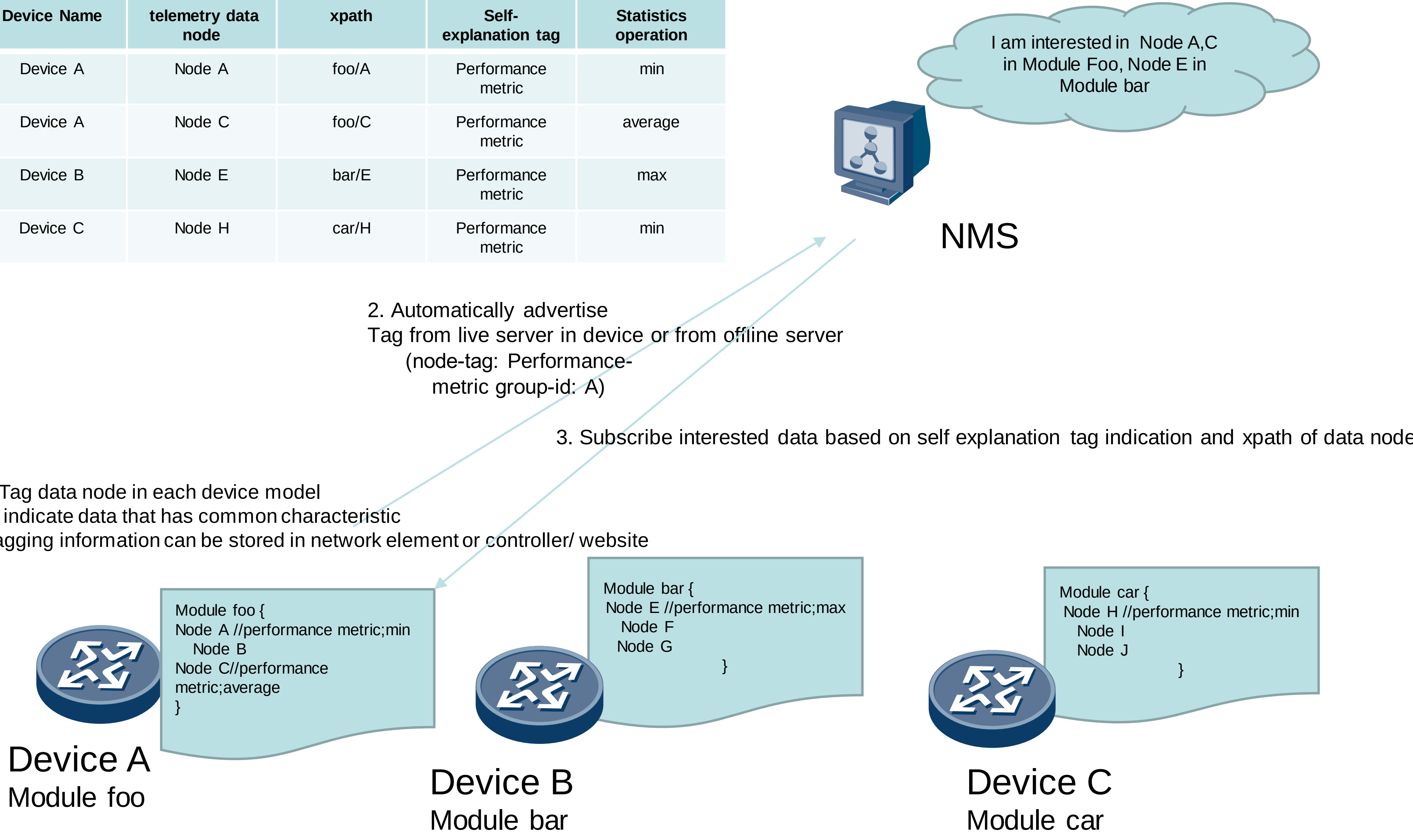
ietf-notification-capabilities Module

ietf-notification-node-tag-capabilities Module

- This document provide a dynamic way to specify selection filter
  - Tag characteristics data object and instruct the client or subscriber to select operation state data based on characteristics data object.
- Data Node Tag Capability Model provides self explanation information for data objects that are of interests.
- Augment Notification Capability model [I-D.netconf-notification-capabilities]
  - specification of which object type nodes they can push to the target recipient.
  - specification of which group of data nodes they can push to the target recipient.
- These self explanation information for data objects can be used to
  - further filter queries of operational state on a server based on server capabilities
  - Correlate data node across models that share common characteristics or are of same object types.
  - reduce the amount of data to be streamed out to the destination.

Data Node Tag Capability Example

Device Name	telemetry data node	xpath	Self-explanation tag	Statistics operation
Device A	Node A	foo/A	Performance metric	min
Device A	Node C	foo/C	Performance metric	average
Device B	Node E	bar/E	Performance metric	max
Device C	Node H	car/H	Performance metric	min



I am interested in Node A,C in Module Foo, Node E in Module bar

2. Automatically advertise
Tag from live server in device or from offline server
(node-tag: Performance-metric group-id: A)

3. Subscribe interested data based on self explanation tag indication and xpath of data node

1.Tag data node in each device model
to indicate data that has common characteristic
Tagging information can be stored in network element or controller/ website

```
Module foo {
Node A //performance metric;min
   Node B
Node C//performance
metric;average
}
```

```
Module bar {
 Node E //performance metric;max
   Node F
  Node G
                }
```

```
Module car {
 Node H //performance metric;min
   Node I
   Node J
                }
```

Next Step

- Decide whether this needs to be documented as RFC by NETCONF WG
- Any comments and input are welcome!