IETF 94 - NMRG 38 meeting
Intent Based Network Modeling (IBNEMO)

Bert Wijnen
Gap between APP and Network

- **APP wants**
  - A connection between two sites
  - A service flow with SLA
  - A customer network service chain

- **APP doesn’t want**
  - Configurations of each device between two sites
  - Flow entries in each device
  - Tunnel, ACL, PBR configurations of different devices
Network application/user use NEMO Language to programming their service

NEMO Language is an Intent oriented network DSL (domain specific language), which is a language style network open interface. Operator/End-user or 3rd party can use it to program network resource and behavior in their service applications.

NEMO Engine is a network middleware, which translate high level service intent to real network instruction base on MDA(Model Driven Architecture).

NEMO is now an OpenDaylight project coming with Beryllium release. https://wiki.opendaylight.org/view/NEMO:Main
Use case: Virtual Private Cloud

- Operator provide virtual private cloud service for enterprises.
  - Need to allocate two zones in the VPC for security issue.
  - DMZ provide http/email/video services for access from internet.
  - Interior provide compute and storage resources for existing enterprise site.
  - Provide BoD capability on the WAN connection.
    - E.g., increase bandwidth when cloud bursting or backup
Use case: End to End Carrier Network

- The enterprise outsources their CPE business to the service provider.
  - The Branch site applies chained services in the vCPE before accessing WAN.
  - The enterprise requests BoD in WAN between the branch and the headquarter for different bandwidth requirements, e.g., day/work and night/backup.
- The NEMO programmed logic runs on the Super controller and instruct domain controller to accomplish network service provisioning and policy applying.
An Example: Bandwidth on Demand (B2B) DOCSIS

• There is a virtual link between the branch and headquarters offices.
  ○ The bandwidth of the vlink can be adjusted on demand
  ○ The adjustment can be triggered by "conditions" meet, e.g. The bandwidth will be adjusted when the timing meets.

NEMO Script:
Node branch;
Node HQ;
Connection tunnel Type P2P
Endnodes branch, HQ;

Constraint day ApplyTo tunnel
  Condition time>8am & time <6pm
  Operation set:bandwidth=10G;
Constraint night ApplyTo tunnel
  Condition(time>0am & time<8am) | (time>6pm & time<0am)
  Operation set:bandwidth=1G;
NEMO Editor in Eclipse

- Program a Eclipse plug-in that highlights the keywords
- Integrate with the IBNEMO project
  - Parse the NEMO language and compose the NEMO rest API.
  - Call rest client/lib to send the request.
- Reference:
- Hackathon implementation result: [https://github.com/paaguti-work/NEMO-IETF](https://github.com/paaguti-work/NEMO-IETF)
What could be the role of intent-based networking in situation where the system's goal "emerges" from the agents interactions?

- **IBNEMO can dynamically update the network behaviour.**
  - So if IBNEMO app gets alert that change is needed, it can do so
  - Tianran Zhou can demo (see him. Possibly also demo at Bits&Bytes)
- **Question: should IBNEMO also monitor network?**
  - Should that be part of IBNEMO?
  - Can it rely on other network monitoring tools that trigger IBNEMO activity?
  - Are there researchers that want to experiment with that?
  - Do we have ideas on how to do this?
What we are looking for: do you have experience in ...

- **operating a network?**
  - How would you implement the use cases?
  - Feedback on the NEMO language definition, proposals for enhancements

- **the OpenDaylight project?**
  - Help us debug the current OpenDaylight NEMO Engine

- **programming plugins for Eclipse?**
  - Program the interface(s) to drive the OpenDaylight implementation
  - Integrate them with the NEMO Editor
  - You know Xtext?
    - Help us to improve the NEMO Editor plugin
Thank You