

Programmatic Interface to Routing

NANOG 61

Draft version, slides will be updated before presentation

Applications and Networks

- **Routing system players: the Application and the Network.**
 - Different interdependent entities and layers
 - Application as seen from Network: it is a payload
 - Network as seen from Application: it is a dumb pipe
- **Application needs Network visibility**
- **Network needs Application awareness**
- **The topic is rapidly evolving and may be controversial**
- **There are multiple concepts in different stages of development in various SDOs. The focus of this talk is on I2RS working group deliverables in IETF.**

Scope

- It is about routing – control plane
- Not about forwarding – data plane
- A router is not necessary an application hosting platform
- Running routing protocol adjacency to an application is not necessary practical
- Reuse existing operational model

Current Operational Model

- **CLI and SNMP are dominant provisioning (configuration) mechanisms**
- **NETCONF is gaining traction as a new provisioning mechanism**
- **But not everything is a configuration – state needs to be modified dynamically and ephemerally**
- **Operational state and statistics needs to be exported**

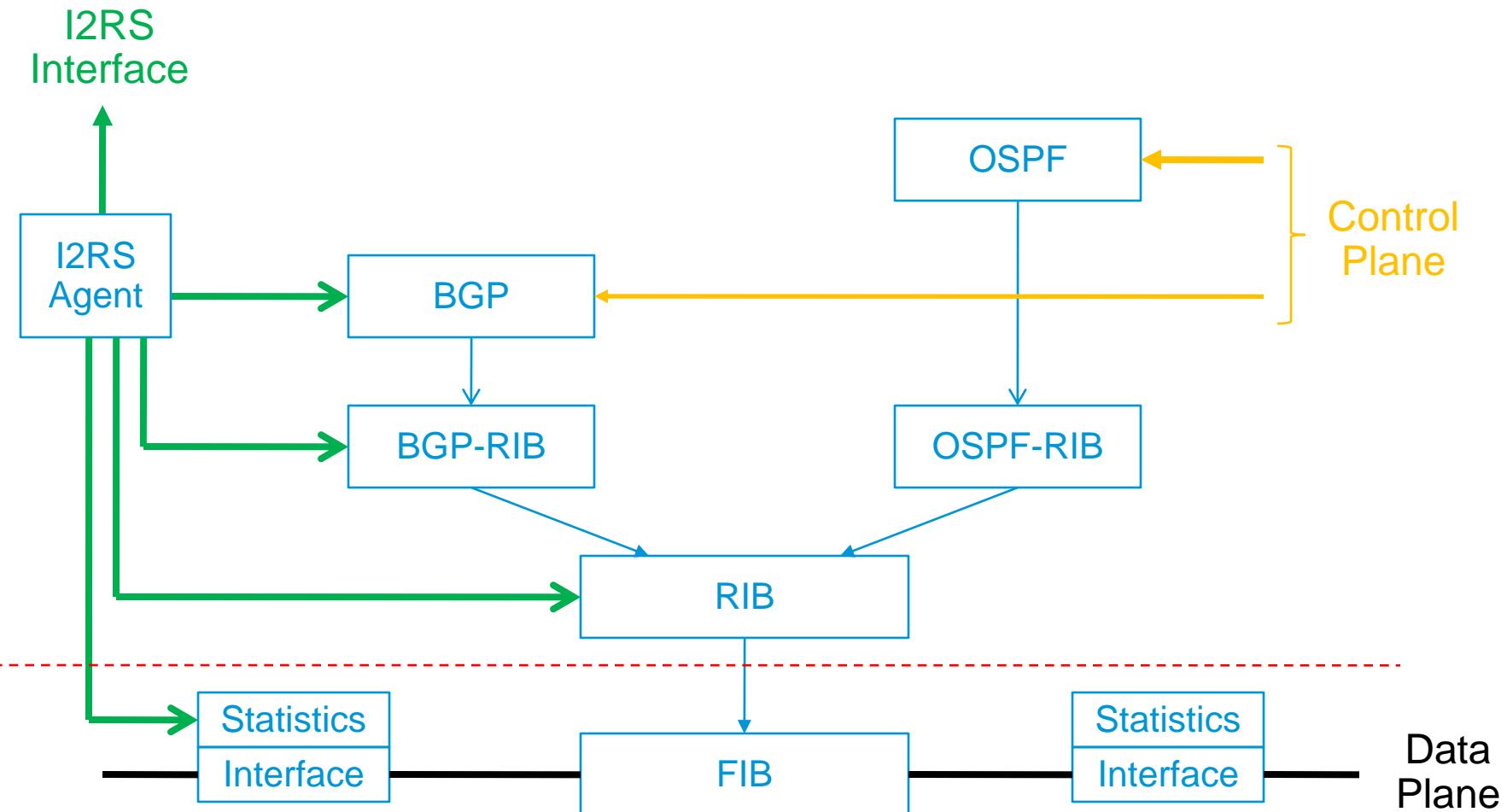
Interfaces and Models

- **Two components – signalling protocol and data model**
- **Bits on the wire, encoding, messaging, state transfer**
- **Description of entities and their attributes**

Forwarding and Platform Aspects

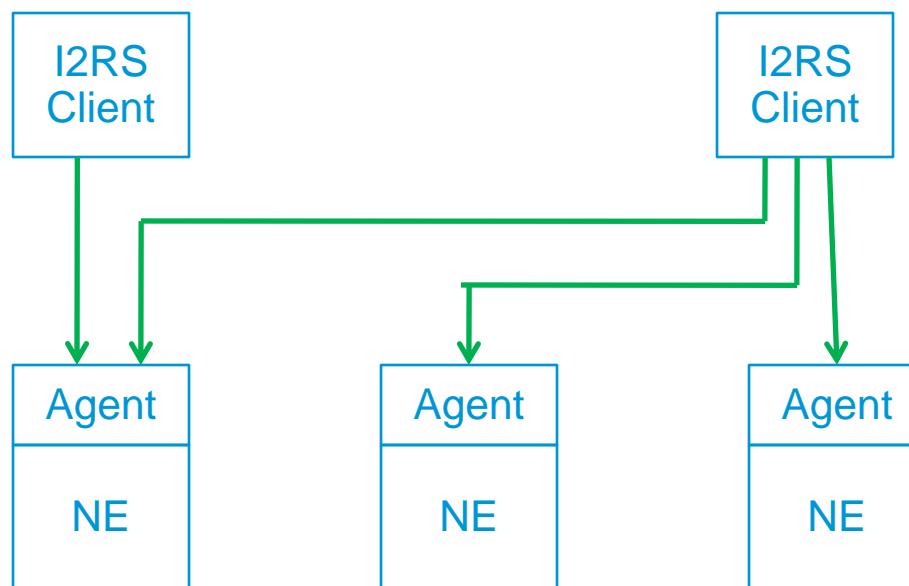
- Reuse existing platforms – do not touch data plane
- Export visibility of data plane operations and events
- Protocol-RIB -> RIB -> FIB model stays the same
- Protocol-RIB changes are (mostly) local
- RIB changes are local
- No changes to FIB

Network Element Model



Network Model

- Clients interact with agents, client to client communication is out of scope of I2RS architecture



I2RS Operational Model Assumptions

- Try to reuse existing operational model
- Collect and export topology data
- Do the application logic part
- Push state into RIBs
- Let platform do the FIB part
- Feed events back to application
- Control plane is still there – and likely unchanged

Use Cases

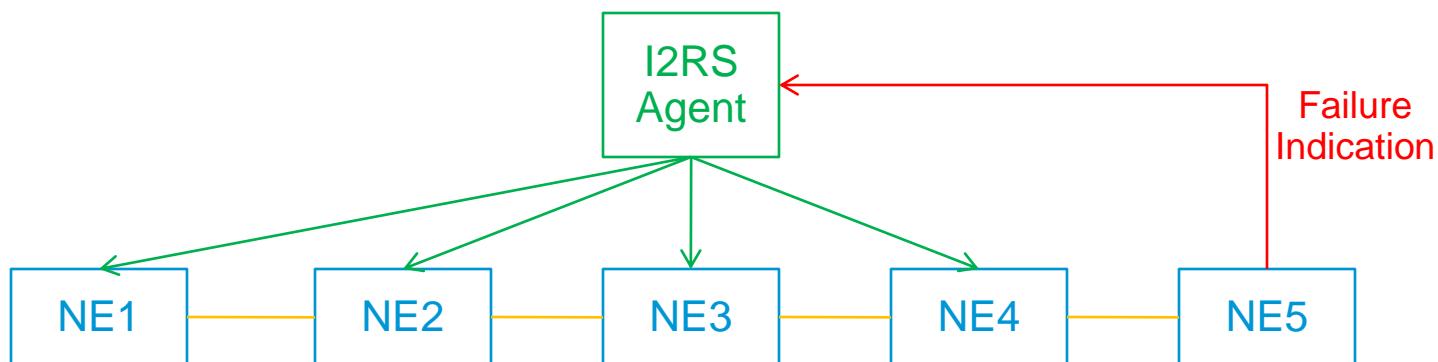
- RIB modifications
- BGP policy augmentation
- Exit path selection
- Topology aware applications
- Protection
- DDoS
- Dynamic overlay topologies
- Centralised control plane operations
- Transport aware routing decisions
- Asynchronous notifications from network layers
- OAM

Use Case – IXP Data Plane Liveness

- In an IXP environment with route server, BGP sessions do not follow the same data plane path as traffic between a pair of IXP peers. Tracking BGP session data plane liveness is not enough.
- Received UPDATE contains new NH attribute
- If path is selected for use, asynchronously notify OAM/data plane liveless component
- Start OAM/data plane liveness session to specified NH node
- Typically solved via automation today, I2RS allows for protocol level automation

Use Case – Failure Notification

- IGP propagates topology changes sequentially neighbor to neighbor, each such hop adds up to the total convergence time.
- The egress node adjacent to failure exports an event "Prefix unreachable" to I2RS client which gets directly propagated to ingress node.
- Ingress node removes affected prefix from RIB and this may trigger restoration process.
- Can be solved via (centralized) event collection today.



Use Case – BGP Policy Augmentation

- Influence BGP best path decision process based on criteria external to BGP attributes
- Per-client custom reflector views
- Centralized attribute management
- Export BGP and BGP-RIB events – new peer, new prefix, change of NH, change of community
- Signal interest to (BGP)-RIB to track state of certain prefixes
- Precomputed BGP RIB download, actual BGP decision process is no longer local.

Use Case – Network Topology Export

- Export information about particular routing protocol topology view without running protocol adjacency to the application
- Export information about RIB topology (similar to TE topology but for routing information only)
- Exported information is implicitly correlated to inventory and operational state
- Potentially in the future topology could be modified via I2RS mechanism too

Use Case – Dynamic Transport

- Large flow transport layer optimization
- Detect large flows – by any practical means
- Lookup flow to forwarding association mapping
- Export flow mapping to transport layer
- Typically solved by offline capacity planning or automation

The Components

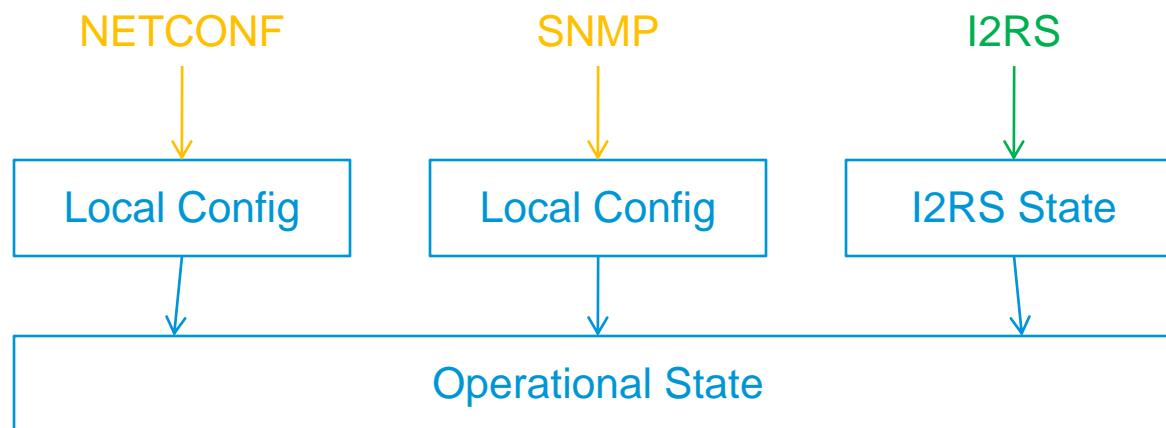
- The overall concept relies on defining a data model of the information that can be exported and imported, and a protocol between I2RS agent and I2RS client
- Protocol part is relatively easy.
- Data model and modelling language is the harder part.
- Both are needed and are interrelated.

The Components

- NETCONF seems to be a growing trend in network element configuration. Naturally it is tempting to reuse it as much as possible.
- Just that NETCONF is inherently tied to configuration, and not so much to operational state that does not have underlying configuration – extensions will be needed.
- ForCES is another option
- YANG with extenstions is a candidate for data modelling language.
- This is an ongoing discussion in the working group at the moment.

Configuration and Operational State

- State received via I2RS interface complements operational state derived from local configuration.



Persistence and Propagation of I2RS State

- I2RS state is defined to be ephemeral, it is not saved into configuration.
- There is no correlation between operational state and configuration state.
- Ephemeral state is injected over I2RS protocol (not yet defined), configuration state is provisioned over NETCONF (or something else – not in scope in this context).
- I2RS state can be propagated to other network elements via normal control plane operation (if injected prefix is considered to be best, BGP will propagate it to other peers).

Discussion

- Ideas on the topic have been floating around in the IETF for a couple of years
- IETF needs community feedback
- Both on architecture and on use cases
- Let's talk