

Notes for I2RS Protocol Design Team meeting on 8/28/2015

Attendees:

- Hariharan Ananthakrishnan hari@packetdesign.com,
- Ignas Bagdonas <ibagdona.ietf@gmail.com>
- Jie Dong (Huawei)
- Susan Hares (Huawei)
- Jan Medved (Cisco)
- Eric Voit (Cisco)
- Kent Watsen (Juniper)

Prior to the meeting, Ignas indicated the BGP use case for I2RS had some changes needed.

Discussion Part 1:

Sue: Alia described the key things for the requirements as being:

- 1) Feedback look for applications,
- 2) Being able to tie ephemeral to config

This discussion is what happens if we consider just these two points. Ephemeral state can tie to:

- Ephemeral to Config (I2RS Route to interface configuration),
- Ephemeral to operational state (Example is LSP-ID) , and
- Ephemeral to ephemeral.

Note: the following point from Jan is important to keep track of:

- Jan: The primary reason for I2RS is the speed of programming. It needs to be faster than configurations, with no internal validation.

Discussion on ephemeral states:

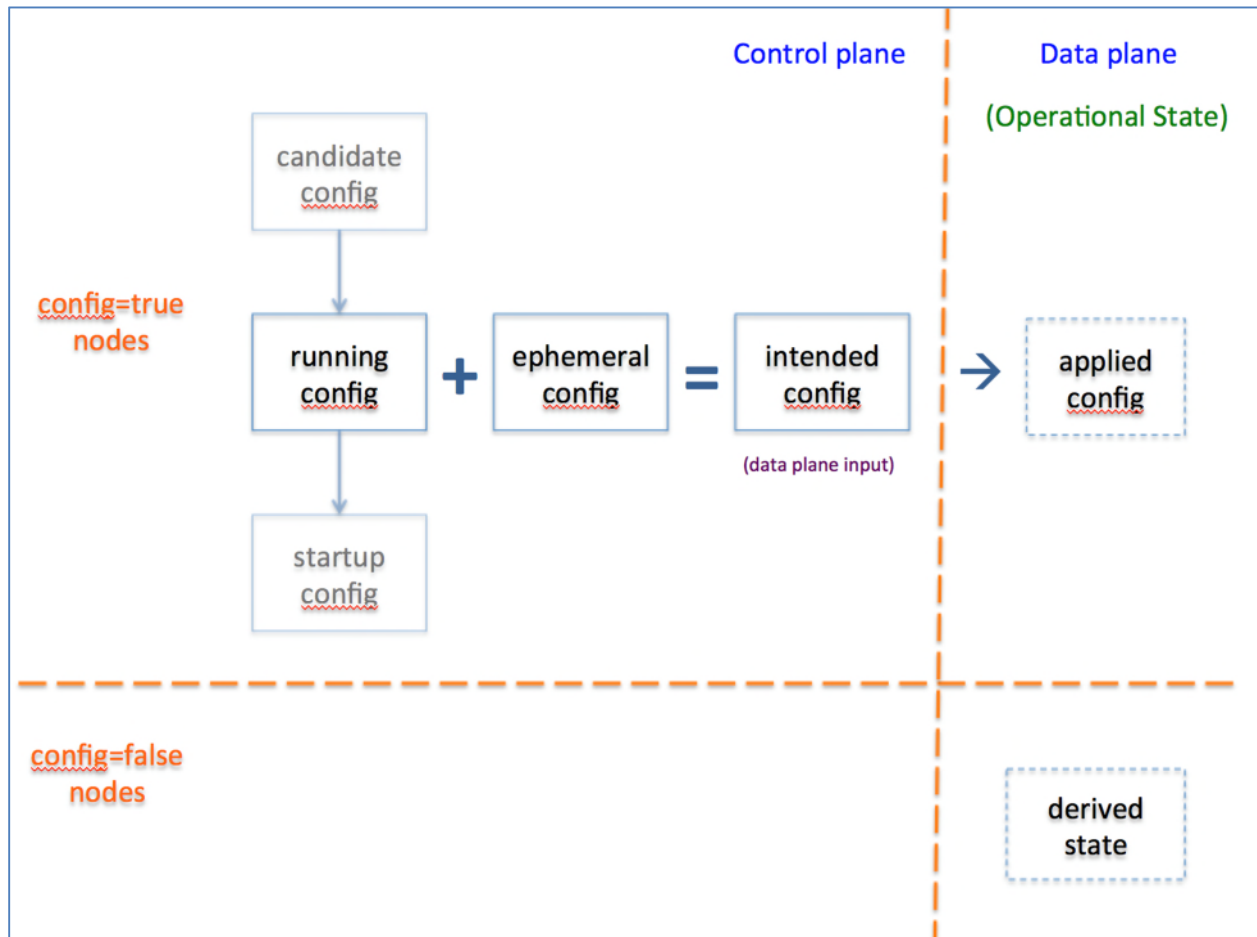
- Hari: If I create some ephemeral nodes, and two agents can both try to write
- Ignas: If one client goes away, the state gets marked. The client gets to know this data.
- Sue: There are two different cases: clients overwriting and the ephemeral state writing within the

Kent: The transitive configuration is flattened. And then transitive configuration comes and get all the information. How it gets from the data plane.

Jan: It is similar. Do we want to use? What performance are we going to get with the current mechanism? How will I get better performance for the I2rs state.

Discussion part 2:

We used the following picture from Kent Watsen to discuss the ephemeral state:



Discussion:

Ignas: Are we talking about Ephemeral state or ephemeral configurations? If we can configure the prefix or we can inject the prefix?

Jan: You can create ephemeral state by inserting ephemeral config.

Ignas: You can consider the LSP case. There is no configuration of the tunnel.

Jan: I would very happy to solve the problem of ephemeral configuration and make it go. The ephemeral to operations is beyond the basic.

Kent: The ephemeral state is a derived state. An example is LSP-ID. It is there derived state.

Ignas: They may be the same.

Kent: The ephemeral state is the same thing as derived state.

Ignas: You will need the same two rectangles in the lower state. One for ephemeral and one for configuration.

Sue: Think of BGP received route with a community configured to it.

Jan: The applied configuration will then associated with the received route.

Sue: yes, if the route goes invalid what is important is that the ephemeral configuration gets notified that it's configuration is invalid.

Jan: You may move from the applied configuration back to the ephemeral state.

Sue: You can keep the ephemeral configuration there as long as you notified it that it no longer applies to real route.

Jan: I agree -You can keep the ephemeral configuration.

Kent: Intended Config is a copy of the dataplane. OpenConfig thinks that the

- Applied = intended configuration + HW
- It can be reduced by hardware.

Sue: Do we believe we can have:

- Ephemeral to config
- Ephemeral to operational state
- Ephemeral to ephemeral

Kent: Derived state is a misnomer?

Jan: What about the performance? If we keep the applied configurations?

Andy: The performance is related to the rules we make to the bells and whistle? The 3 ways to have all or none?

Jan: Bells/whistle vs. performance – Let's stay with making the performance better.

Andy: I test it against a super-simple model of thermostat. The actual is 50 degrees. Tell the systems is 120 degrees. The injection of state is applied configuration.

Jan: We've only create a configuration model. The use case for setting a Community on model. We created BGP yang models for the ODL?

Sue: Are you changing the BGP control plane with ODL BGP yang modules?

Jan: The yang modules focus on put lots of routes into the BGP table. This is covering most of people's needs.

Jan: People are trying to call and this is running.

Kent: We can design an ephemeral configuration. Something an I2RS simple

Jan: One of the use cases is programming a forwarder with I2RS. The requirement is to program 50K routes per second. You do not directorate. You have to be very fast. We are using a simple use case. Do this really quick? We use plane NETCONF and do 50K routes. We do have persistent turned on. We are looking to push 150K routes at a time. Sometime, we have parallelization of route pushes.

Sue: I agree this is a good approach.

Jan: Let's do the speed and encryption. When we take open-flow it is a lot slower. It is a lot slower because it cannot do batching. We need to make sure.

Ken: Simplified yang model – what would be the running configuration, intent configuration, applied, derived?

Jan: We could try the following:

- a) Simplified yang model – what would be running config, intent configuration, applied, Derived state.
- b) Instances of something ACL or RIB model or L2 MAC dress, then see what times to scale.

Kent: I am looking at the model, and you are looking at the protocol model. Are you looking at protocol?

Andy: You are looking at the co-cores that are looking at 16 sessions and they are all sending routes concurrently. The agent is processing read/writes concurrently. It can lead to good performance or segment faults.

Jan: We need to look at the scale use case after we look at the thermostat. That is the crawling?

Sue: Do you want to start with something simple RIB?

Kent: If we keep the model small – 100 lines. We can give it a shot. Everyone know how to use a temperature model. Other people do not understand. What the thing is suppose to do.

Jan: If you have a complicated of RIB model, you will use a very small subset. It can be compressed into a 100 yang lines.

Sue: I could take a crack at the 100 line Yang model.

Jan: Most of the time you use the basic model. That can you. Crawling with thermostat and model, and then walking with 100 line yang model?

Sue: How about Andy and Kent working out the thermostat model, and I'll work through the I2RS simple RIB model with input from Hari and Jie Dong for Next week. Hari implemented the RIB model with NETCONF so he's got a good idea.

Kent: This sounds good.

- 1) Time zero – ephemeral config
- 2) Intent config –
- 3) Applied config –
- 4) Plus derived state.

Sue – I can get the RIB state through the work, and to show the sequence. I think I can get this done by Monday so we can discuss the viewpoints.

Kent: I would show the NETCONF command that would target thermostat.

Sue: Sounds like a good plan for 9/4/2015 meeting.