Server to ToR signaling

draft-kompella-nvo3-server2tor

Kireeti Kompella - Yakov Rekhter - **Thomas Morin**

Context

- The focus of this presentation is the case where the NVE is on the ToR device
  - ie. virtual network connectivity provided by ToR devices

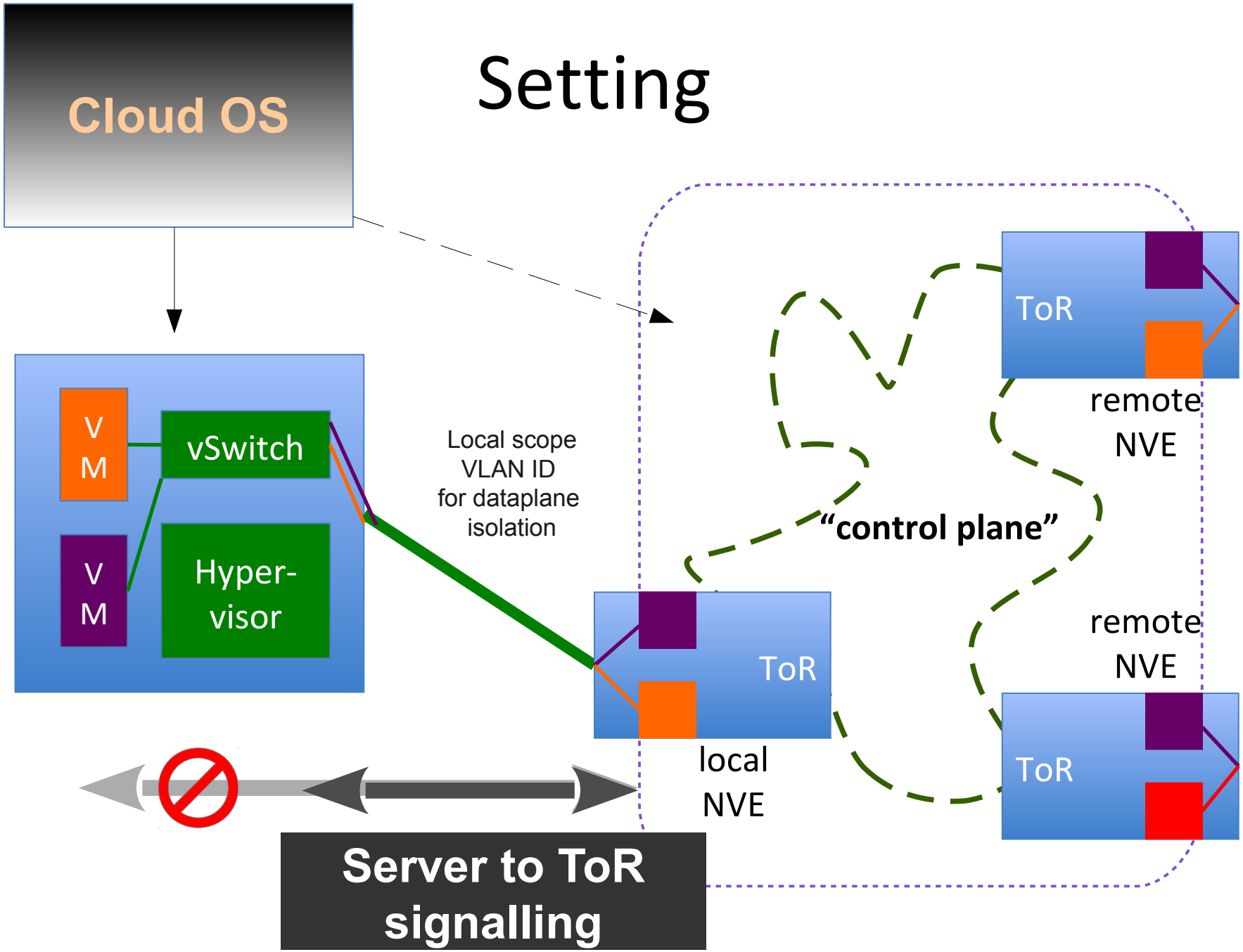
Goals of this talk

- Describe the goals of this signaling
  - what, not how
- Describe the setting for this signaling
  - dispel some of the confusion here
- Say how this fits in with other mechanisms
  - “Control plane”, the Cloud OS, …
- Say where we’d like to take the draft
  - Given support from the WG, of course

Goals of server2tor signaling

- Goals:
  1. **Single provisioning touchpoint**
     - (no provisioning of network devices in the critical path)
  2. **No need for Cloud OS to be aware of DC network topology**
     - for virtual network setup and VM mobility
  3. Synchronize parameters between server and NVE (local scope VLAN id)

Setting


Cloud OS
V M
V M
vSwitch
Hyper-visor
Local scope VLAN ID for dataplane isolation
ToR
remote NVE
"control plane"
ToR
local NVE
remote NVE
ToR
Server to ToR signalling

Signaling : what ? when ?

- What is exchanged:
  - information letting the server determine which local VLAN id to use for a said VM
- When does signaling happen:
  - at VM instantiation time
  - at VM migration time
  - at VM termination time

Genericity

- Need for a generic solution:
  - agnostic to hypervisors
    - (esp. how a said hypervisor handles VM migration)
  - agnostic to whatever solution is used between ToRs :
    - 'pick your poison' : TRILL, E-VPN, IP VPN, LISP, SPB, proprietary XYZ, etc...
- Need to have extensibility:
  - different ways to identify a virtual network
  - different mux/demux ?

Authorization

- In this approach, the server is acting on behalf of the Cloud OS to trigger virtual networking
- It seems useful to allow the server to prove that this indeed is the case
- Need to allow the server2tor signaling to carry such a proof

Minimizing traffic loss during VM migration [1/2]

•Setting: a VM is moved from old server S connected to old ToR L to new server S' connected to new ToR L'

•Two distributed events on servers:

- S told to pause VM
- S' told to start VM

•Two distributed events on ToRs:

- L told that VM has paused
- L' told the VM has started

Minimizing traffic loss during VM migration [2/2]

- Cannot assume strict time ordering among these events
- If L knows that VM has been paused locally and that VM has moved to S'/L', L can relay traffic to VM to L'
- This happens when remote NVEs haven't yet got the update that VM moved
- If L' isn't ready, it drops the traffic, but that's no worse than L dropping it

Incremental deployment ? [1/2]

Limitations of ARP (in no particular order) :

- No way to “withdraw” an IP/MAC association
  - ARP relies on timeout
- Hard to distinguish move from multi-homing
- No way to authenticate ARPs
- No way to include a VNID in ARP message
  - unless the VNID *is* the VLAN tag
- If ARP is just for server2tor signaling, ToR has to intercept ARP and translate to control plane
- (possibly more)

Incremental deployment ? [2/2]

- Incremental deployment is nice to have
- We don't believe ARP is a starting point
- Open question : how can this be done ?

Use inside a server

- Main focus of the proposal is signaling between server and ToR switch
- We can conceive that such a signaling could also be useful inside a server, when NVE is on the server :

  use this signaling between the cloud OS agent on the compute node and the vswitch

- Could allow writing a single Cloud OS plugin for NVE which would be usable in both cases of NVE location (on server and on ToR)

Next steps

- We think this is useful
  - need for documenting requirements ?
- We are interested on comments on the approach
  - thanks for those already made !
- There are several possible candidates for formatting the messages
  - and variants on details of information carried in the messages