Deploying a Stateful and Fault Tolerant Virtual Gateway using Open vSwitch in SD-WAN

Sabyasachi Sengupta
Nuage Networks (Nokia)
(sabyasachi.sengupta@nokia.com)

Open vSwitch Fall Conference 2016

Linux Foundation

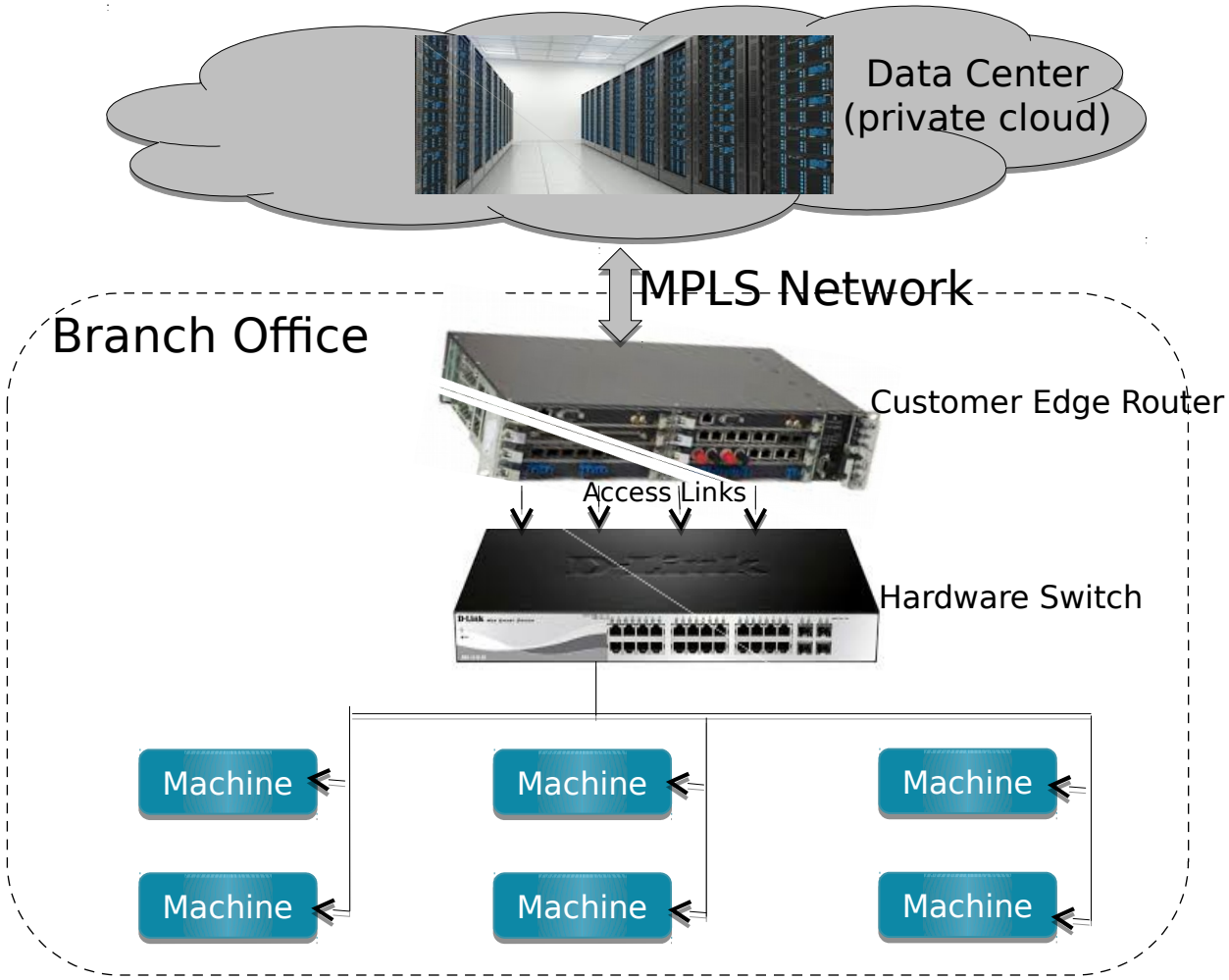
Agenda

- Background
- Problem Statement
- Proposed Solution
- Solution Details
- Conclusion

- Background – 3 mins
  - Brief Introduction about Traditional Branch Routing & SDWAN
- Problem Statement
- Proposed Solution
- Solution Details
- Conclusion

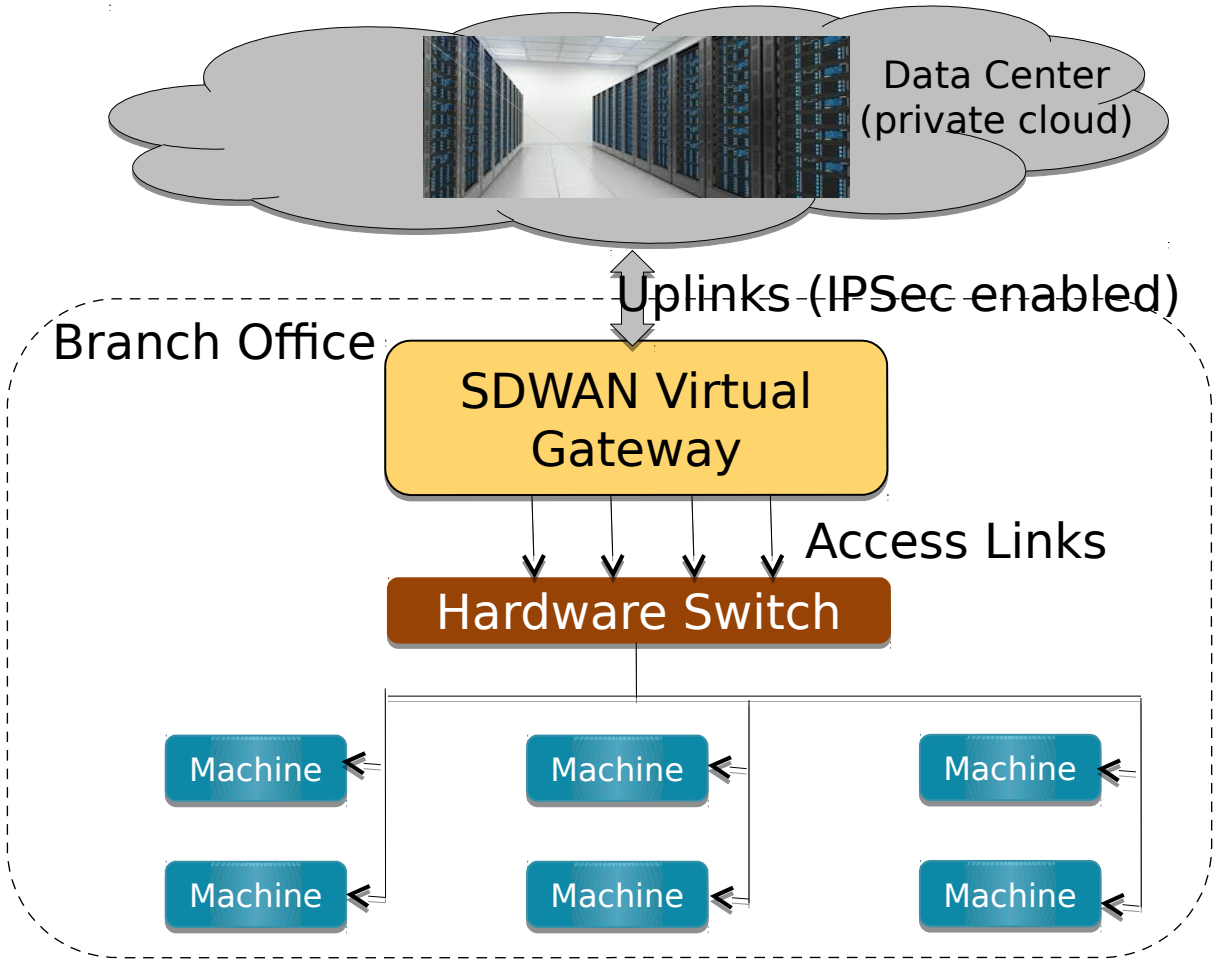
Background

Branch Networking (Oversimplified)



Background (CONT)

Current SDWAN Landscape (oversimplified)



Background (CONT)

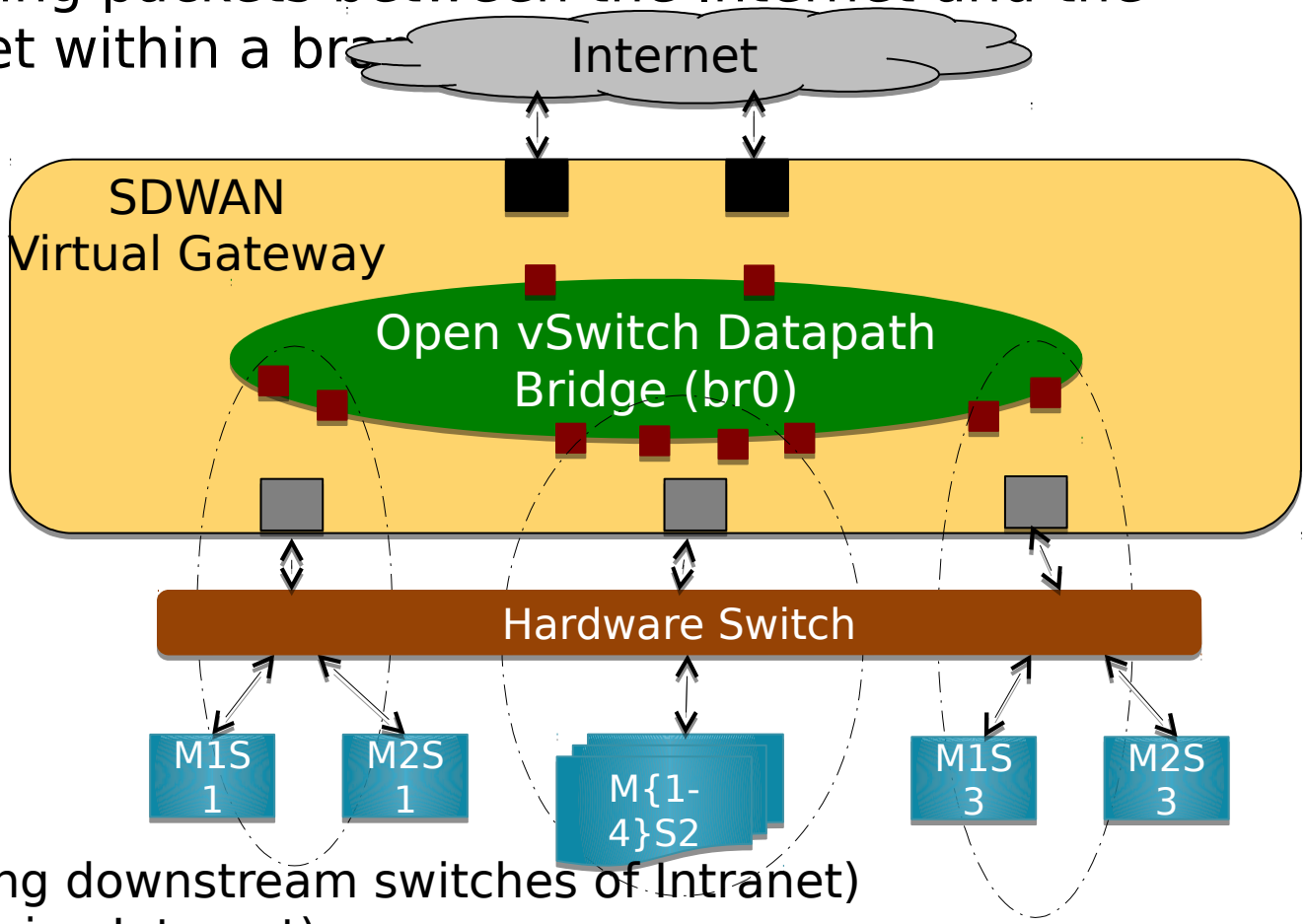
SDWAN Virtual Gateway (SVG)

- In modern day architecture, the *SDWAN Virtual Gateways* usually run a virtual switch along with other virtualized network appliances for meeting networking needs
- Virtual Gateways are built using
  - Off-the-shelf commodity hardware, with
    - One or more *Uplink Ports* facing the Internet
    - Few *Access Ports* facing the downstream switches or machines of the Intranet
  - Customized software
    - Linux Base OS, viz. Redhat
    - Virtualization Software, viz. KVM
    - Virtual Switch, viz. *Open vSwitch*

Background (CONT)

Open vSwitch in SVG

- Switching packets between the Internet and the Intranet within a bra



LEGEND

Access Ports (facing downstream switches of Intranet)

Uplink Ports (facing Internet)

Virtual Ports (vLan Ports created on top of Access Ports for each machine / sub

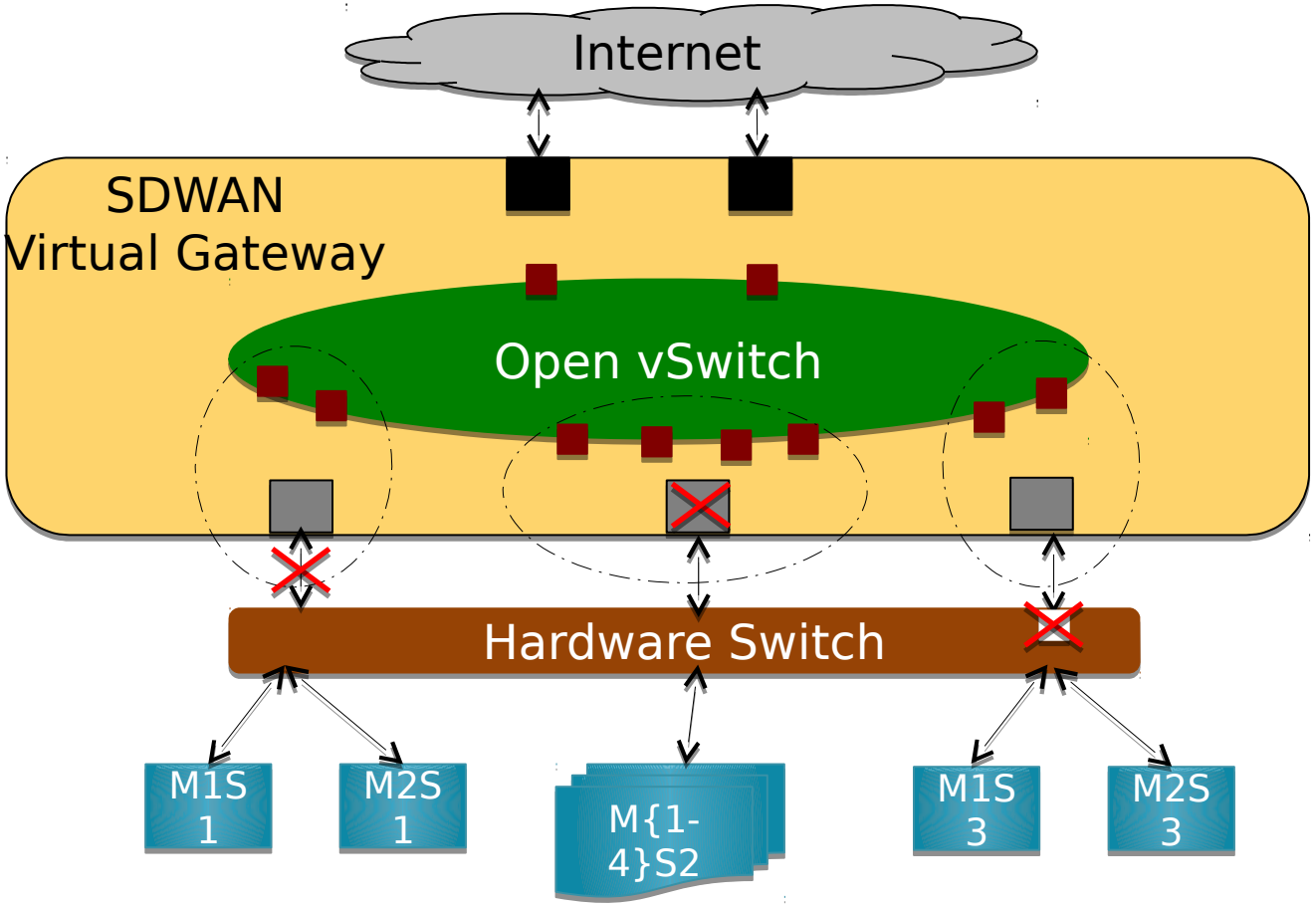
Bridge managed by OVS (br0)

PC or Virtual Machines connected to SVG – MiSj: Machine #i in subnet #j

- Background
- Problem Statement – 2 mins
  - A look at the requirements and various problem scenarios
- Proposed Solution
- Solution Details
- Conclusion

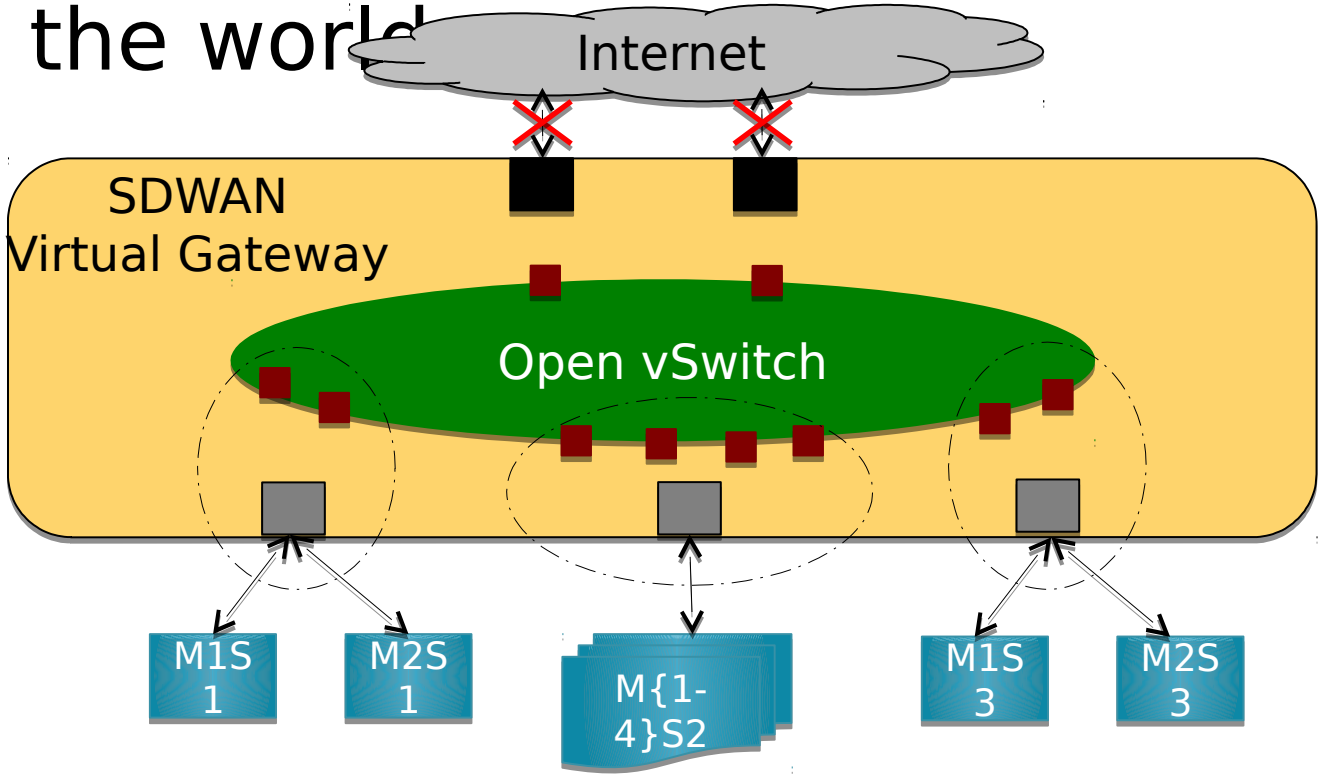
Problem Statement

- Access Links connecting the underlying hardware switch may fail resulting in branch site being disconnected from other sites.



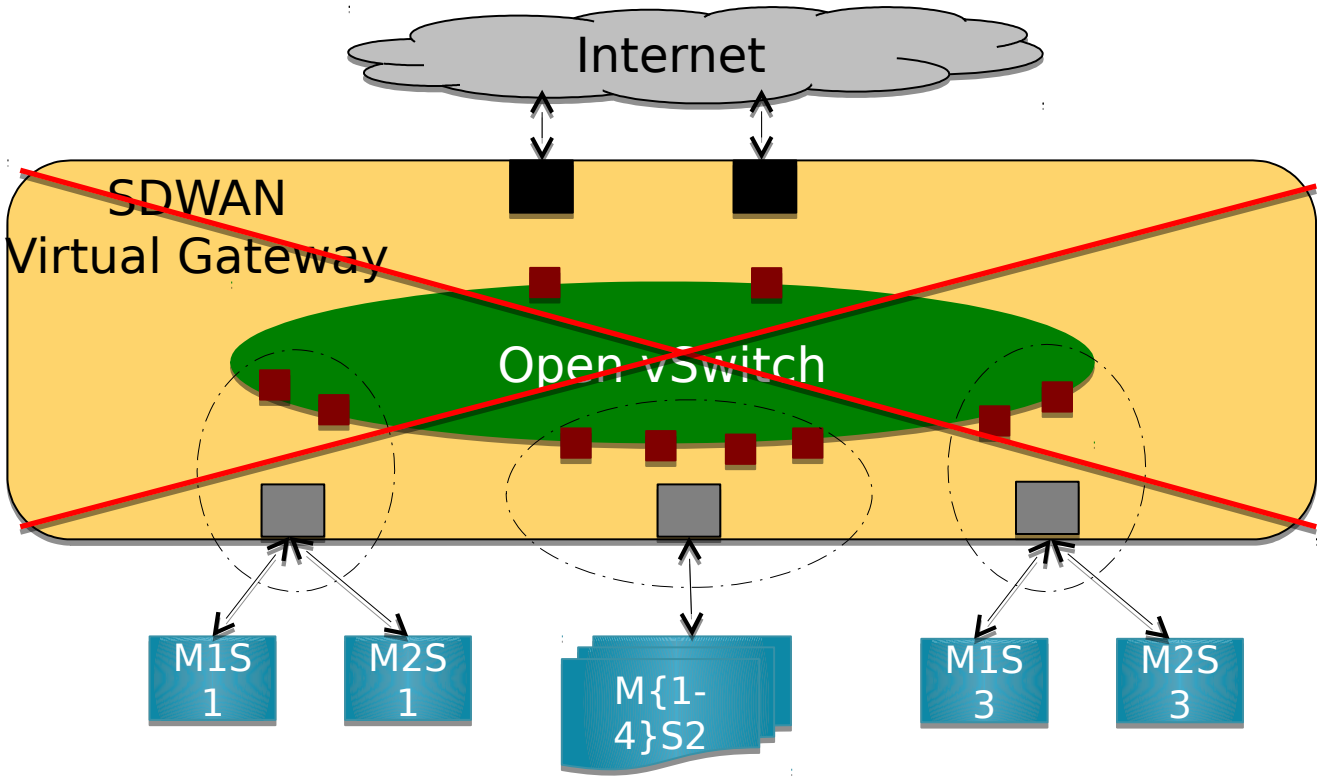
Problem Statement (CONT)

- If all uplinks fail, branch network will be partitioned from rest of the world



Problem Statement (CONT)

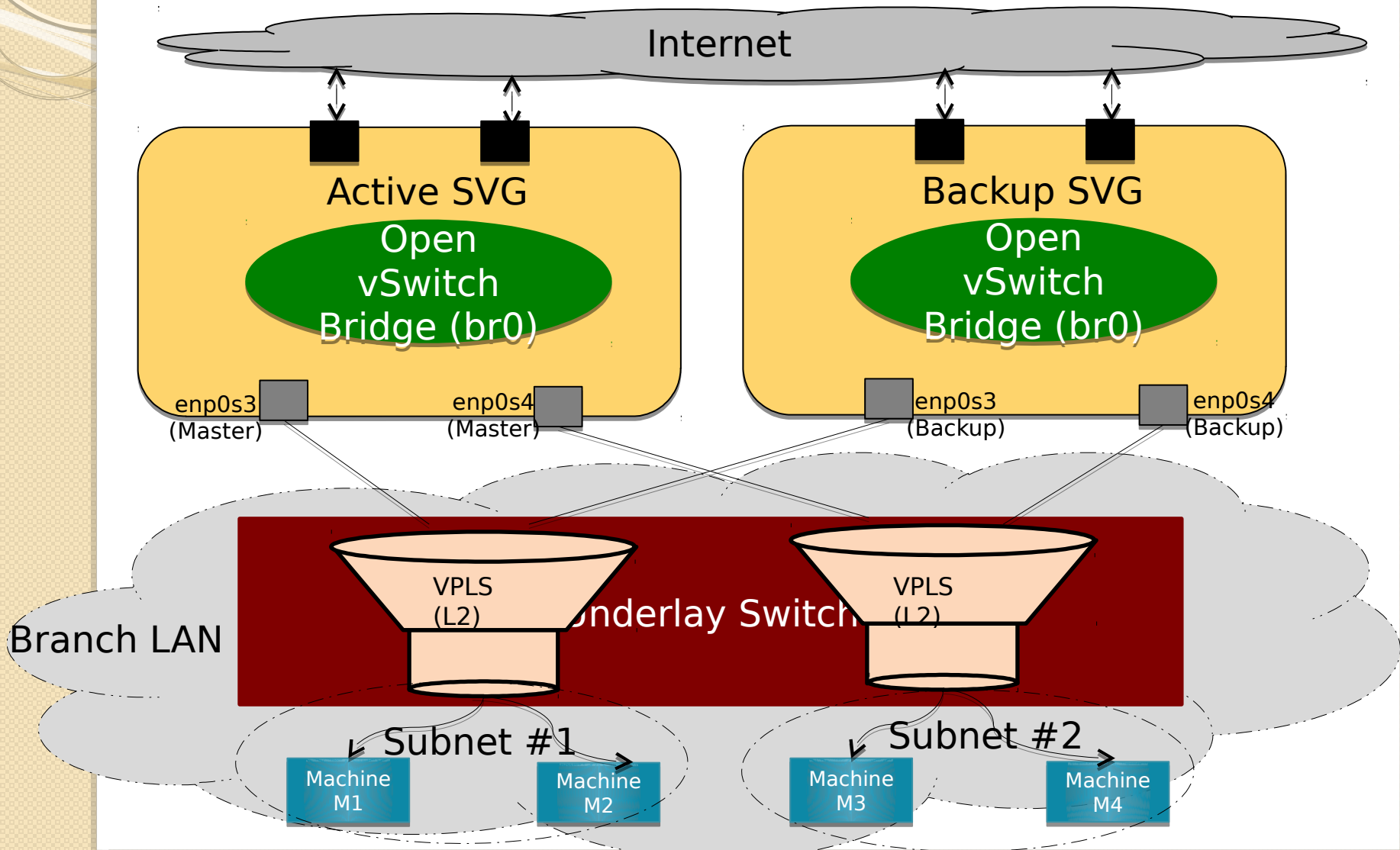
- What if forwarding engine in the SVG crashes?
- What if SVG itself crashes?



- Background
- Problem Statement
- Proposed Solution – 8 mins
  - Key aspects of the solution, BFD extensions and overview of Open vSwitch changes
- Solution Details
- Conclusion

Proposed Solution

Solution Architecture *(Physical View)*



Proposed Solution (CONT)

Explanation

- SVG can be deployed in Active-Passive mode to achieve High Availability
  - *Master SVG* – SVG configured as *active*
  - *Backup SVG* – SVG configured as *backup*
- Each access port can be individually configured as either active or backup depending on the need.
- BFD monitoring can be deployed between access link pairs
  - *Bidirectional Forwarding Decision (BFD)* is a network protocol that detects faults between two forwarding engines connected by a link
  - Open vSwitch has BFD implementation that supports link monitoring
  - However there is no support yet for checking active-passiveness (mastership) of a link

Proposed Solution (CONT)

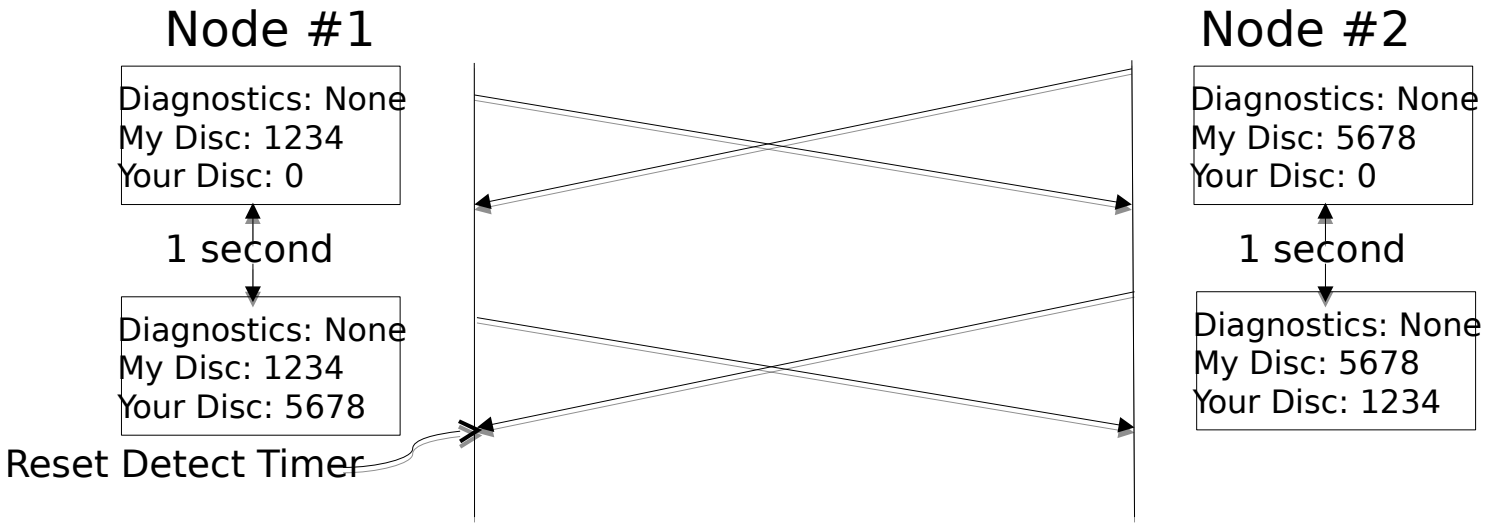
Thus Spake RFC5880 – BFD specifications

BFD Ver (3)	Diag (5)	State (2)	Flags (6)	Mult_detect (8)	Length (8)
My Discriminator (32)					
Your Discriminator (32)					
Desired Minimum TX Interval					
Desired Minimum RX Interval					
Desired Minimum Echo RX Interval					

- My Discriminator:
  - A unique, nonzero discriminator value generated by the transmitting system, used to demultiplex multiple BFD sessions between the same pair of systems.
- Your Discriminator:
  - The discriminator received from the corresponding remote system. This field reflects back the received value of My Discriminator, or is zero if that value is unknown.

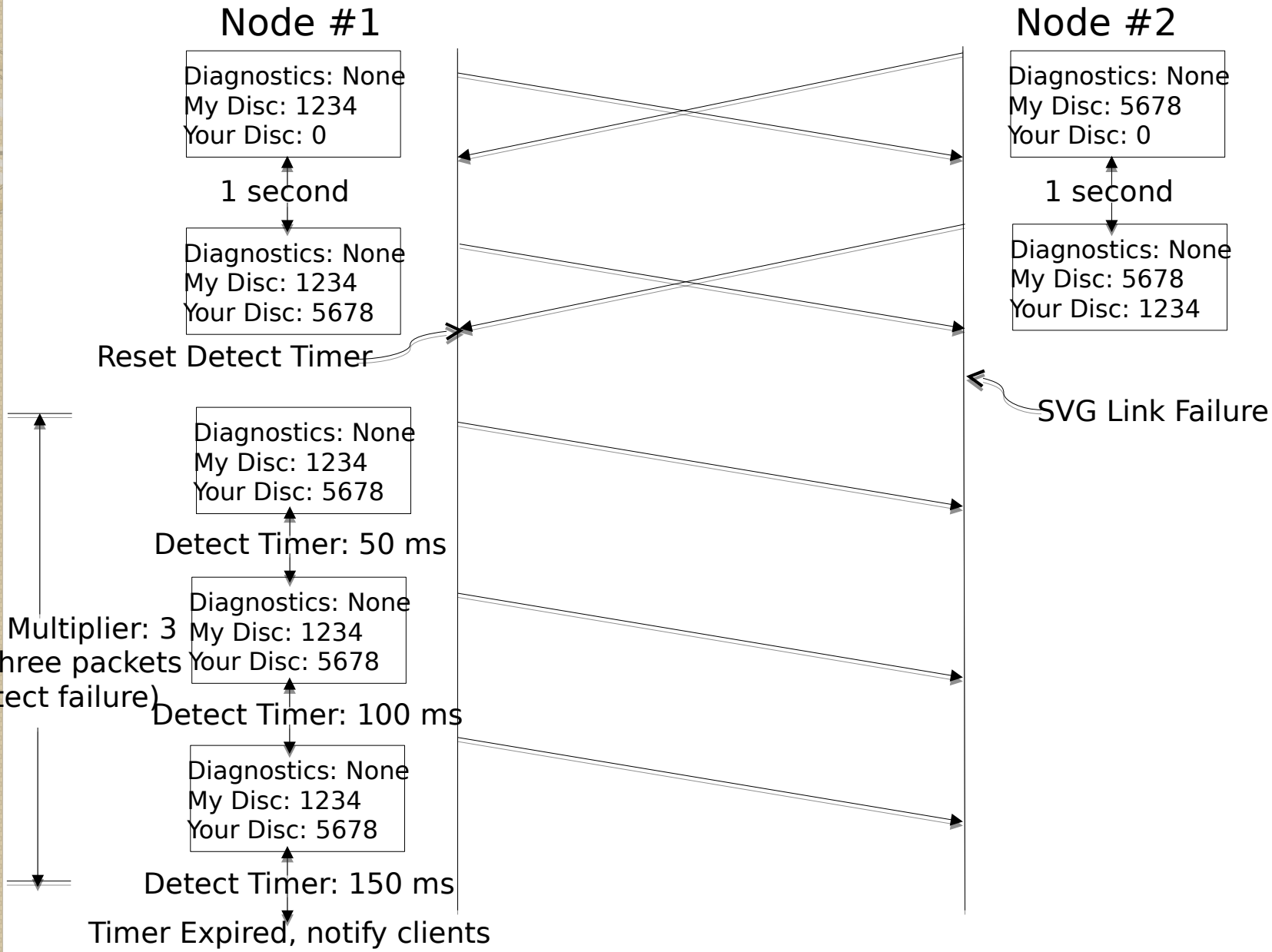
Proposed Solution (CONT)

High Level BFD Operation Timing Diagram – Recap



Proposed Solution (CONT)

High Level BFD Operation Timing Diagram – Recap



Proposed Solution (CONT)

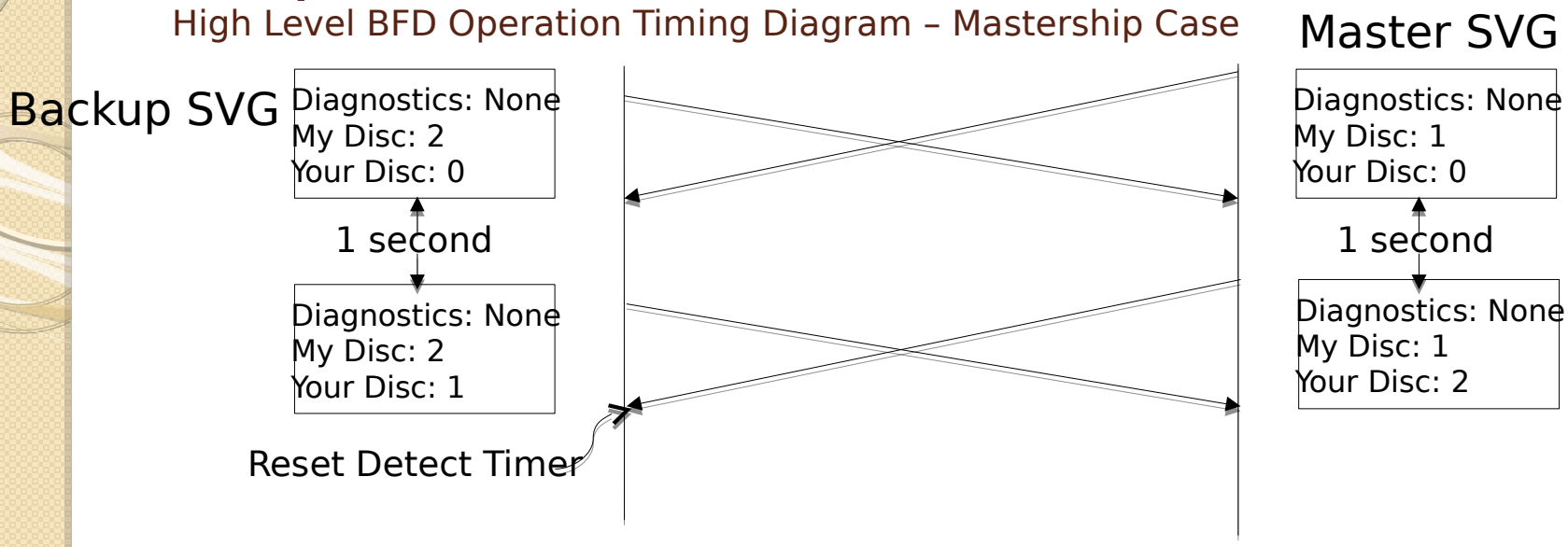
RFC5880 Extension

BFD Ver (3)	Diag (5)	State (2)	Flags (6)	Mult_detect (8)	Length (8)
Local openflow port number			Local mastership role		
Remote openflow port number			Remote mastership role		
Desired Minimum TX Interval					
Desired Minimum RX Interval					
Desired Minimum Echo RX Interval					

- Mastership Role: Can assume two values
  - 1: link is designated as master role
  - 2: link is designated as backup role
- Openflow port number: The openflow port number field can be used as an unique handle that can be passed on to different subsystems to determine on which port the mastership is configured.

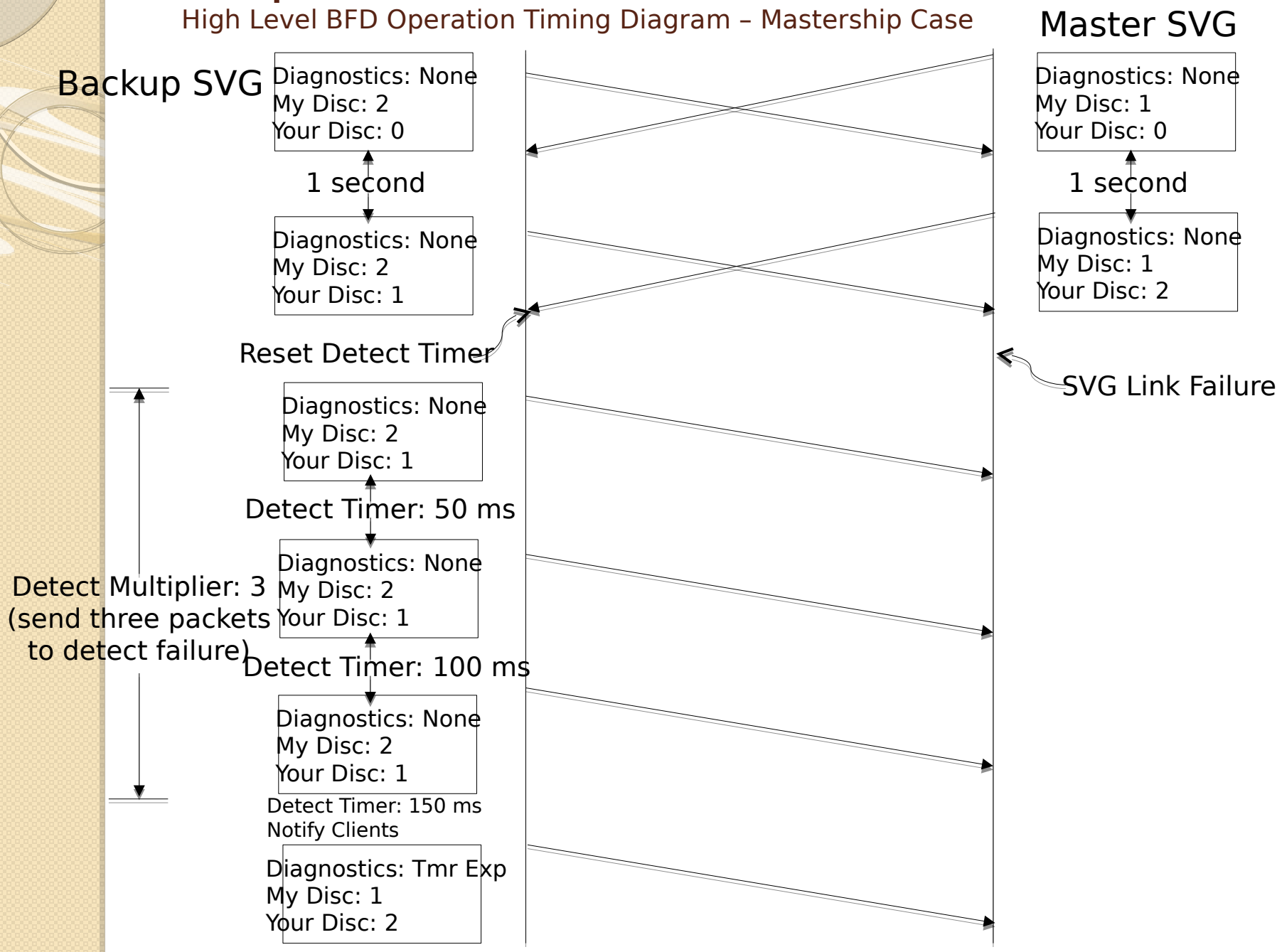
Proposed Solution (CONT)

High Level BFD Operation Timing Diagram – Mastership Case



Proposed Solution (CONT)

High Level BFD Operation Timing Diagram – Mastership Case



Proposed Solution (CONT)

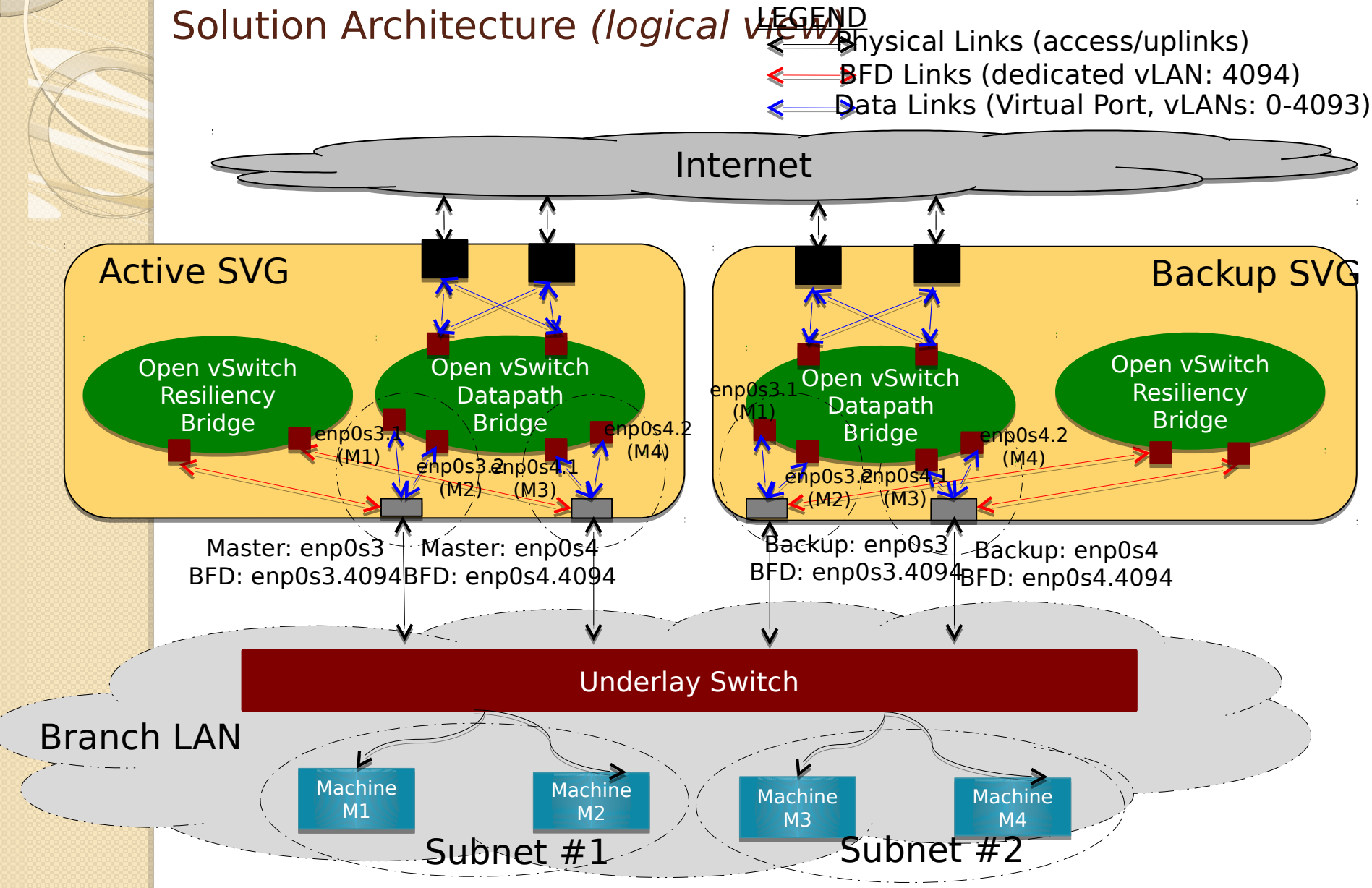
Overview of Modifications in Core OVS – Tested with ovs-2.5

- Configuration through Openflow
  - Added `OFPTYPE_BFD_MASTERSHIP_MOD` openflow command that process BFD mastership configuration from CMS/Controller
  - Modified `handle_openflow__` to modify mastership information and save in *bfd structure*
- Configuration through OVSDB
  - `bfd_configure` – controller programs the configured mastership state in OVSDB. Mastership configuration is read from OVSDB and saved in *bfd* structure during runtime of virtual switch
- Mastership election
  - `bfd_process_packet` – negotiate mastership with peer SVG node and setup bfd.my_disc and bfd.your_disc appropriately based on state at current node
  - `bfd_run` – setting up current node's BFD mastership state based on link state changes
- Linux Network Stack Interaction
  - `netdev_linux_run` – monitor uplink state and force set local SVG discriminators so that mastership can be negotiated with peer.
- Presentation
  - `bfd_put_details` – Added a couple of extra fields to display local and peer node state in bfd/show (debugability)

- Background
- Problem Statement
- Proposed Solution
- Solution Details – 10 mins
  - Digging deeper into how the SDWAN resiliency solution fits in with quick reference to different use cases
- Conclusion

Solution Details (CONT)

Solution Architecture *(logical view)*



Solution Details (CONT)

Challenges – Dropping traffic at backup SVG

- Issue: Backup SVG must not forward any traffic, else there will be duplicate packets in the network
  - As underlay switch is configured in VPLS, it forms an L2 network containing the active and backup SVG ports and machines from the branch
  - Traffic originating from branch machines, will be sent to the switch and it will send traffic to both SVG ports
  - SVG ports will need to reject traffic arriving at the backup port.
- Solution: Create a rule with *drop* action whenever BFD mastership election flaps to backup.

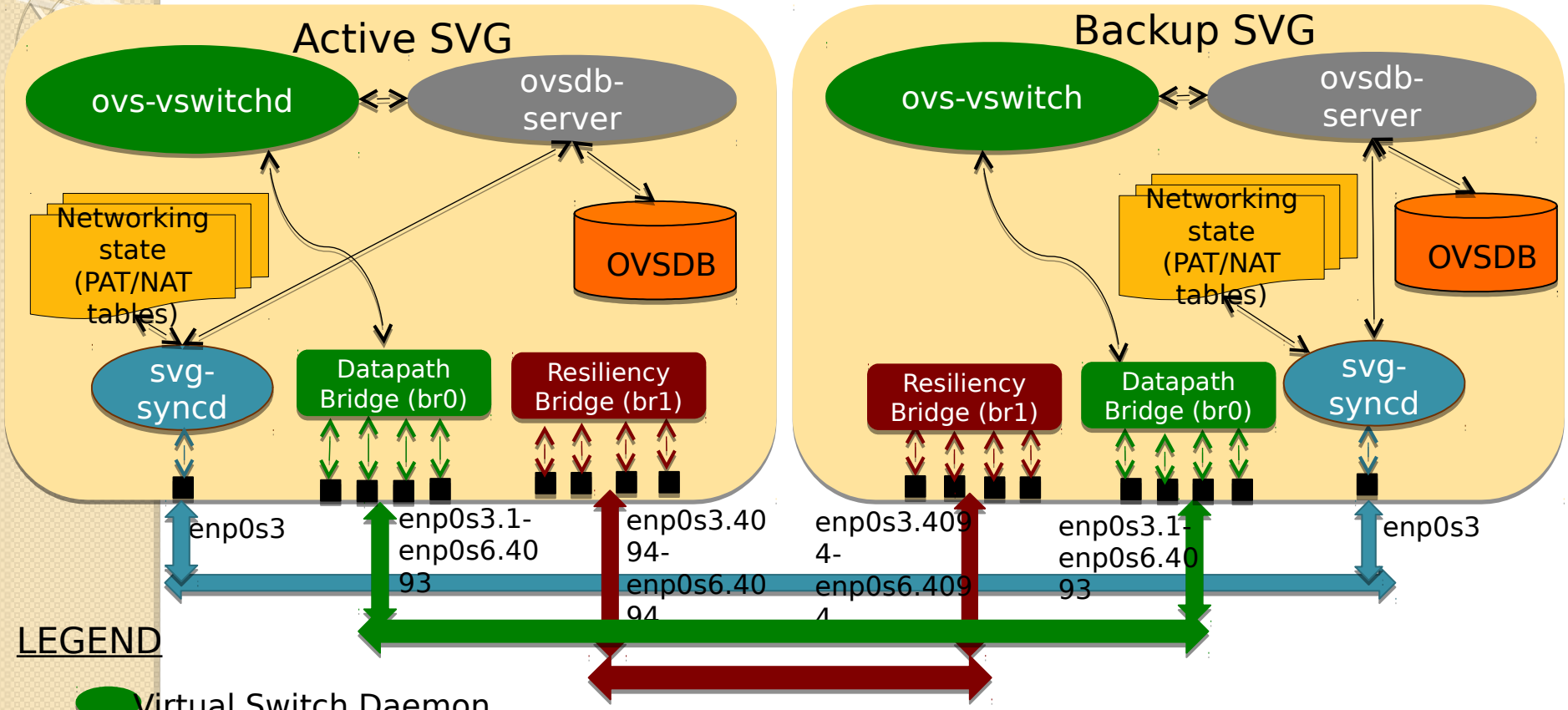
Solution Details (CONT)

Challenges (CONT) – Gateway Statefulness

- Issue #1: With every mastership flap, all the state maintained in the Active SVG must be seamlessly available in the Backup SVG so that it can kickstart all forwarding and other network service provider responsibilities *as soon as possible*
  - Examples: DHCP services for all nodes in the branch, PAT/NAT translations etc.
  - Solution: All Virtual Networking state saved in OVSDB and elsewhere in the system must be transferred to backup and maintained in real time
- Issue #2: It may so happen that few of the access ports can act as *master* in Active SVG, while some others act as backup at the same time. This means that traffic can be forwarded by either of the two SVGs for different subnets at the same time.
  - Solution: State synchronization must be bi-directional between the Active and Backup SVG.

Solution Details (CONT)

Gateway State Synchronization Mechanism



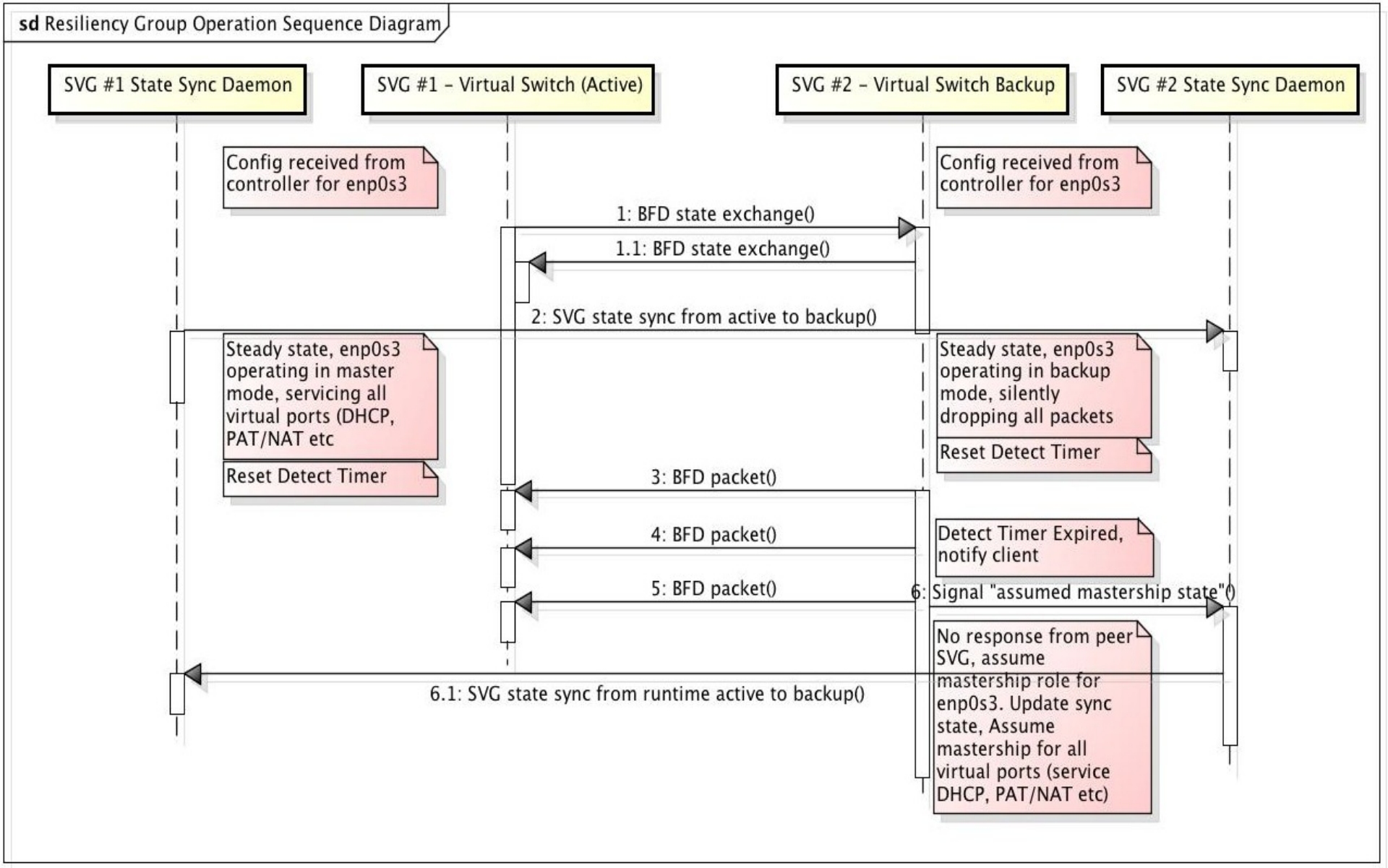
Solution Details (CONT)

Challenges (CONT) – Avoiding *split-brain* situation

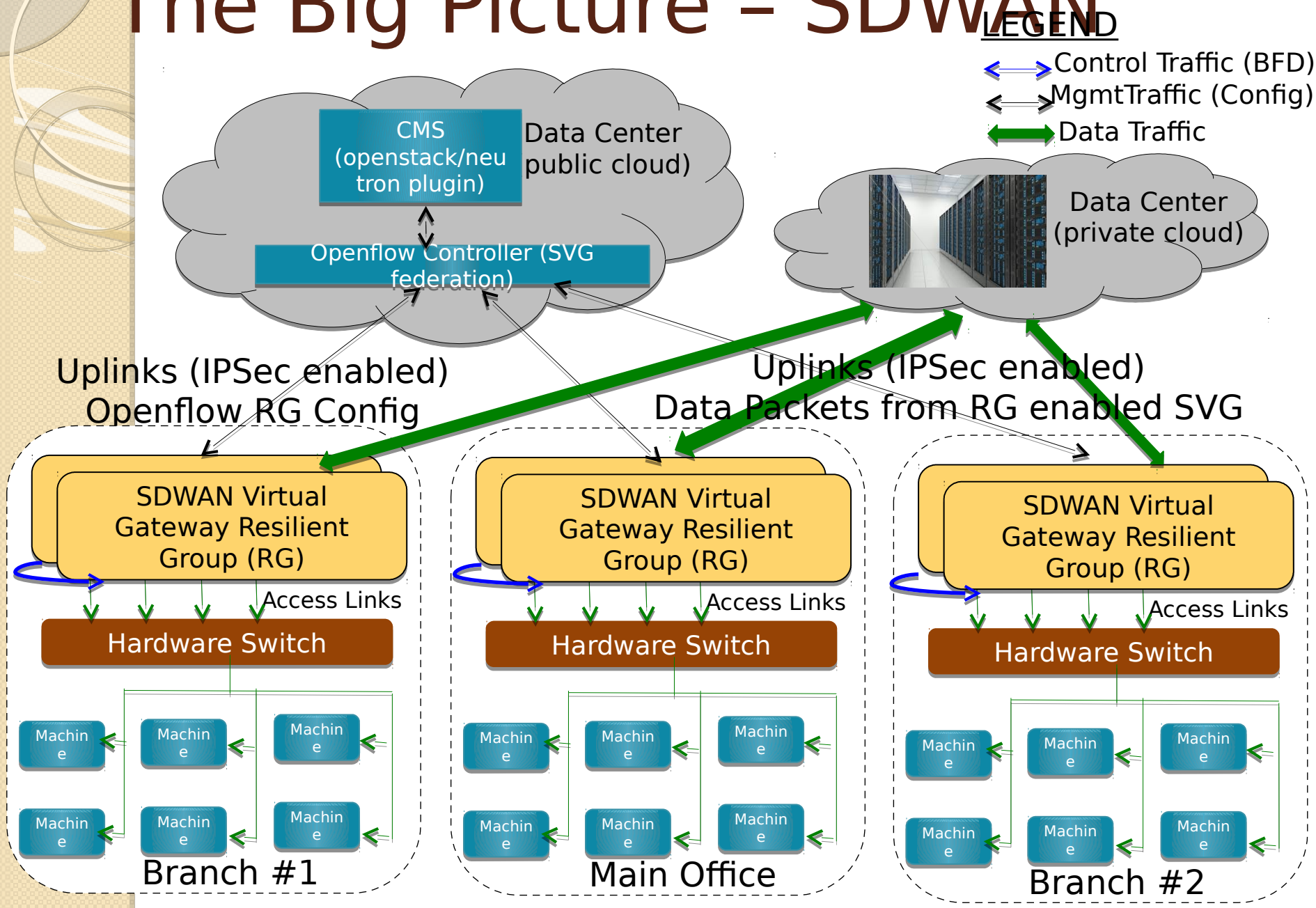
- Issue: Can occur if downstream hardware switch fails.
  - Can result in both SVGs assuming master role and advertise wrong routes thereby attracting traffic only to eventually drop them.
- Solution(s):
  - Recommend customer to use redundant downstream switches
  - Use BFD heartbeats between management ports and refuse a backup port to assume master if mgmt port BFD is dead.

Solution Detail (CONT)

How does it all fit it?



The Big Picture – SDWAN



- Background
- Problem Statement
- Proposed Solution
- Solution Details
- Conclusion – 2 mins
  - Where we are, where do we go from here and references

Conclusions

- In this work, we demonstrated how RFC5880 implementation in Open vSwitch can be extended to do mastership election.
- This coupled with additional features such as synchronizing statefulness can be used to build a fault tolerant and highly available gateway in SDWAN.

Conclusion

- Current Status
  - Most of the work has been modeled in *Virtual Network Services (VNS)* gateway product in Nuage Networks / Nokia
    - Work started in 2015 and fully completed in 2016
    - Shipping since late 2015.
  - Credits
    - PLM: Rotem Solomonovich
    - Developer: Sabyasachi Sengupta (SDWAN device), Natalia Balus (CMS) & Karthik Sankaran (SDN controller)
    - System Test: Mahesh K Thangavel
- Future Work
  - Mastership election in BFD forms the core of this work and can be contributed to Open vSwitch upstream in one of its future releases.
  - Most of the SVG state synchronization can be done using distributed capabilities of OVSDB, however, it may require some enhancements, especially for bidirectional replication and row-level granularity.

Conclusion (CONT)

Reference Materials and Further Reading

- Nuage Networks SDWAN Brochure
  - http://www.nuagenetworks.net/wp-content/uploads/2015/04/PR1503009766_NN_VNS_Extensible_Wide_Area-Networking_Brochure.pdf
- Nuage Virtual Networking Service Access Resiliency Manual (customer login/password required):
  - https://infoproducts.alcatel-lucent.com/aces/htdocs/3HE10719AAAE/VNSGuide-70-access-resiliency.html
- RFC5880 Specifications
  - https://tools.ietf.org/html/rfc5880
- OVSDB Replication (ovs-2.6)
  - https://github.com/openvswitch/ovs/blob/master/Documentation/OVSDB-

Questions?