Congestion Isolation – Design Team Update

Kevin Shen, Paul Congdon, Yolanda Yu (Huawei),

Carmi Arad (Marvell),

Feng Gao (Baidu)

IEEE 802.1 DCB

Orlando, FL

November 2017

Agenda

- Congestion Isolation Refresh
- Changes since last time
- Next Steps

Congestion Isolation

Definition: An approach to isolate flows causing congestion and signal upstream to isolate the same flows to avoid head-of-line blocking.

The approach involves:

1. Identifying the flows creating congestion (e.g. perhaps already done for QCN and/or ECN)
2. Using implementation specific approaches to dynamically adjust the traffic class of offending flows without packet re-ordering (e.g. DVL – Dynamic Virtual Lanes)
3. Signaling upstream indications via a Congestion Isolation Packet (CIP)

Isolate the congestion to mitigate HOLB

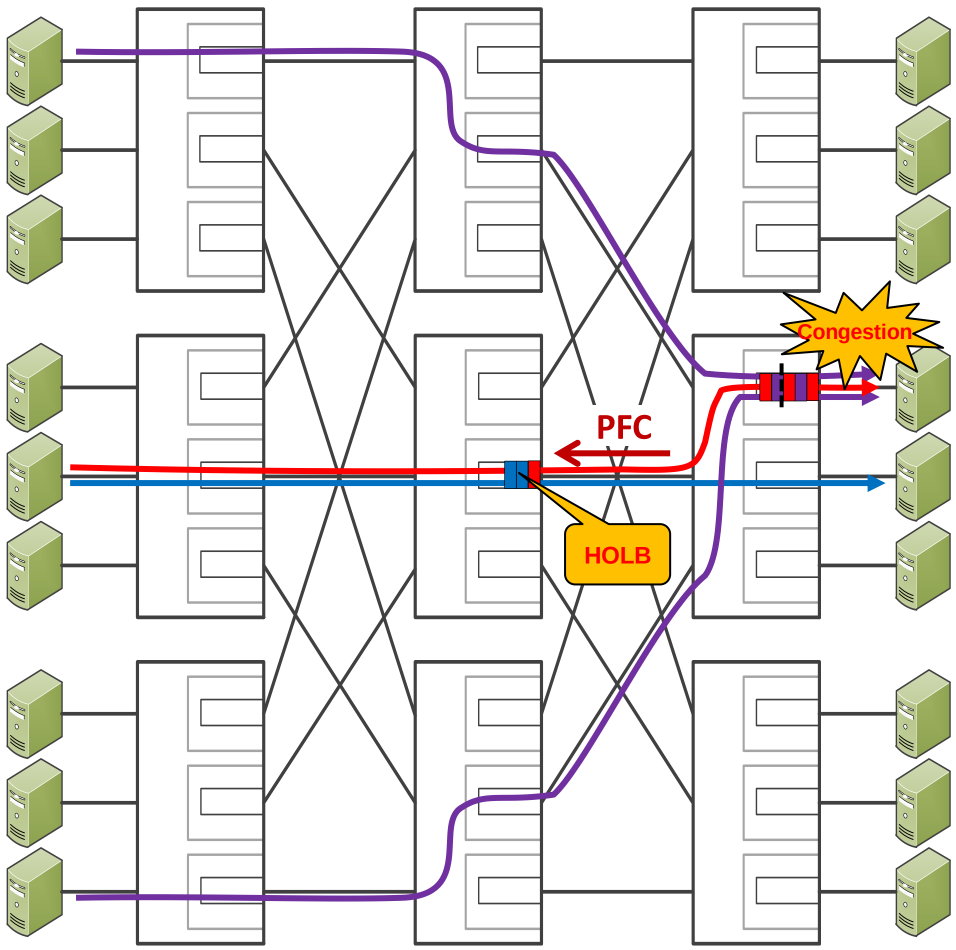
Congested Flow →

Congested Queue 🟩

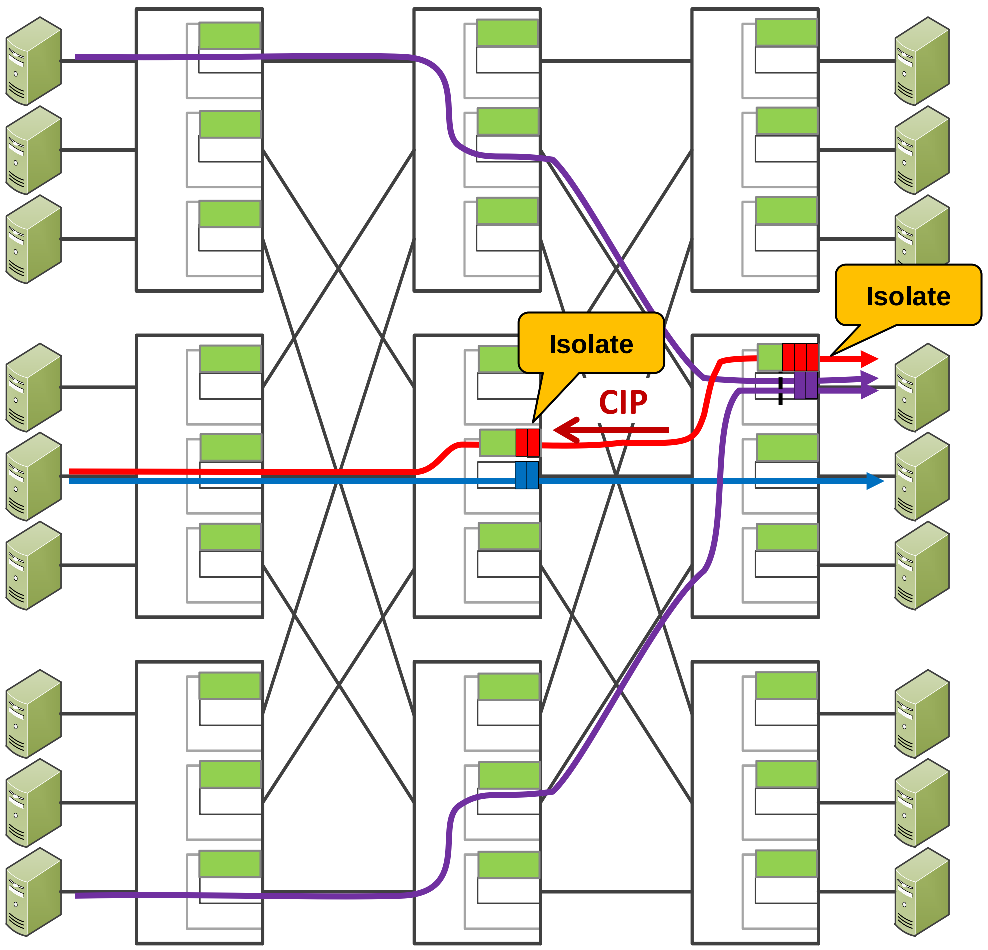
Victim Flow →

Normal Queue □

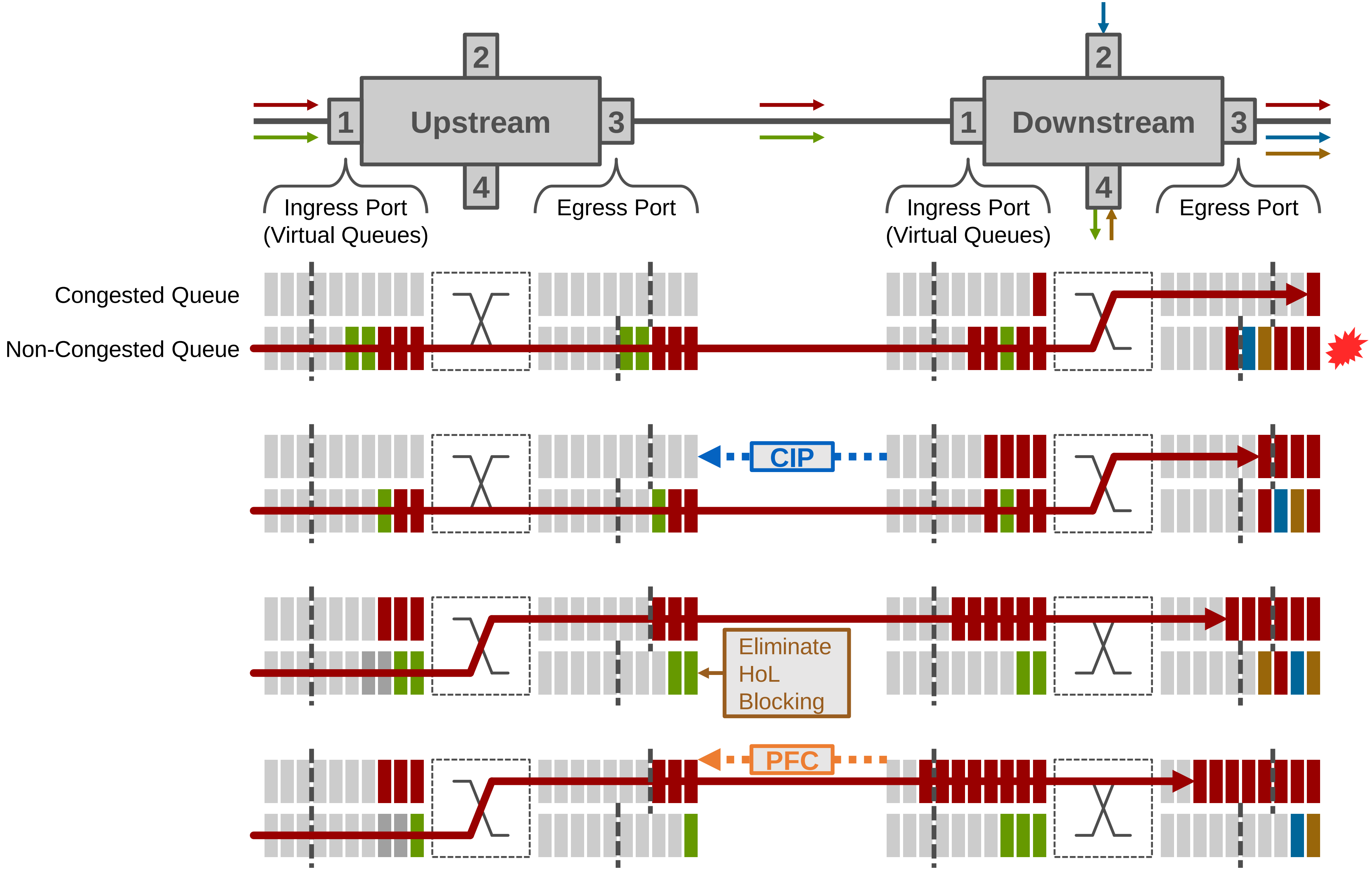
Today – Without Congestion Isolation



Congestion Isolation



Congestion Isolation



1. Identify the flow causing congestion and isolate locally

2. Signal to neighbor when congested queue fills

3. Upstream isolates the flow too, eliminating head-of-line blocking

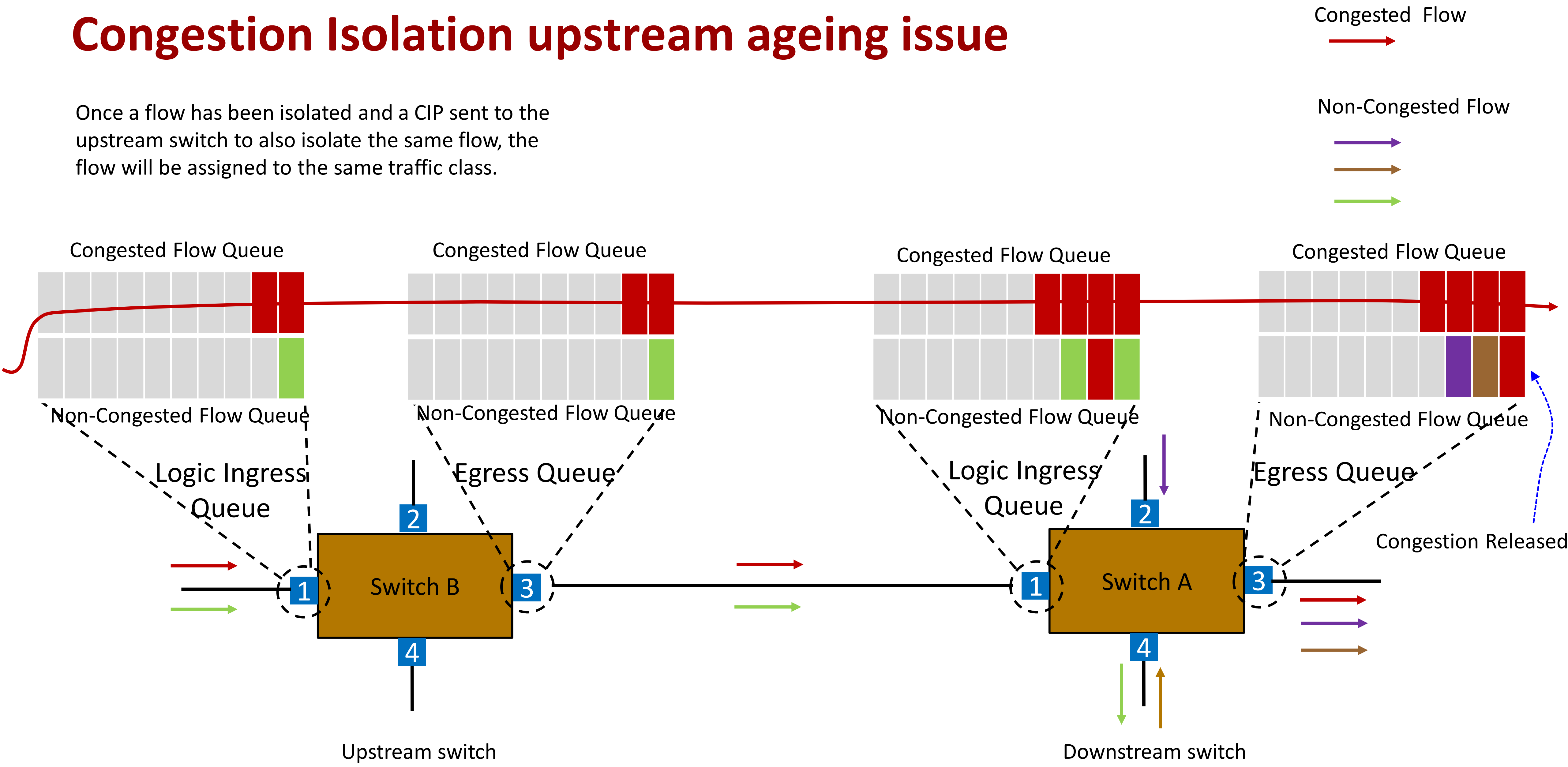
4. If congested queue continues to fill, invoke PFC for lossless

Design team discussion topics and resolution proposals

- Asynchronous upstream ageing
- Re-use 802.1Qau CNM message format
- Specifying order preservation after isolation
- Neighbor capability discovery
- Operation in hierarchical networks (e.g. VXLAN)
- Multicast Operation
- Congested flows changing paths
- Recovery From Loss Of CIP Messages

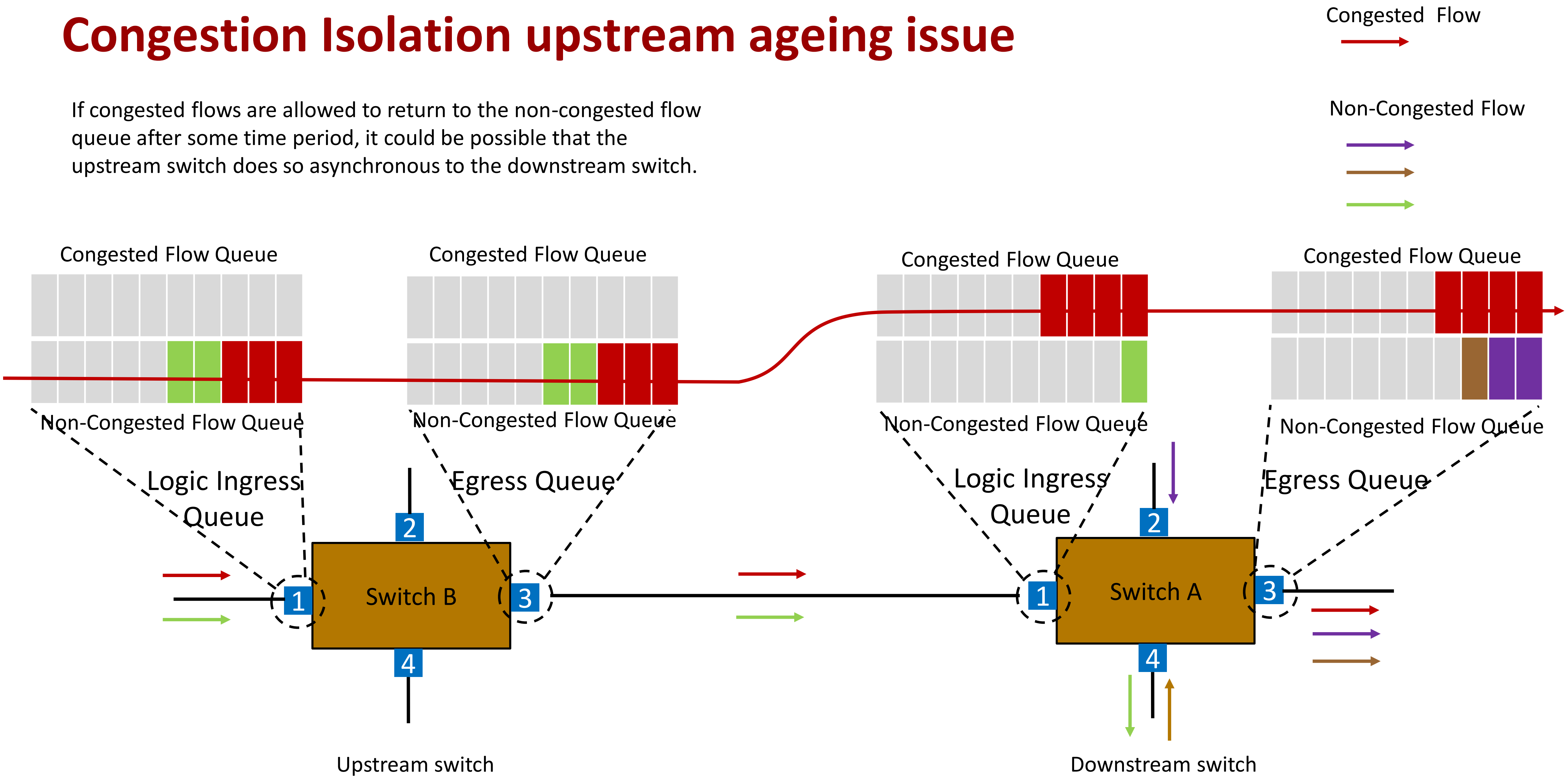
Congestion Isolation upstream ageing issue

Once a flow has been isolated and a CIP sent to the upstream switch to also isolate the same flow, the flow will be assigned to the same traffic class.



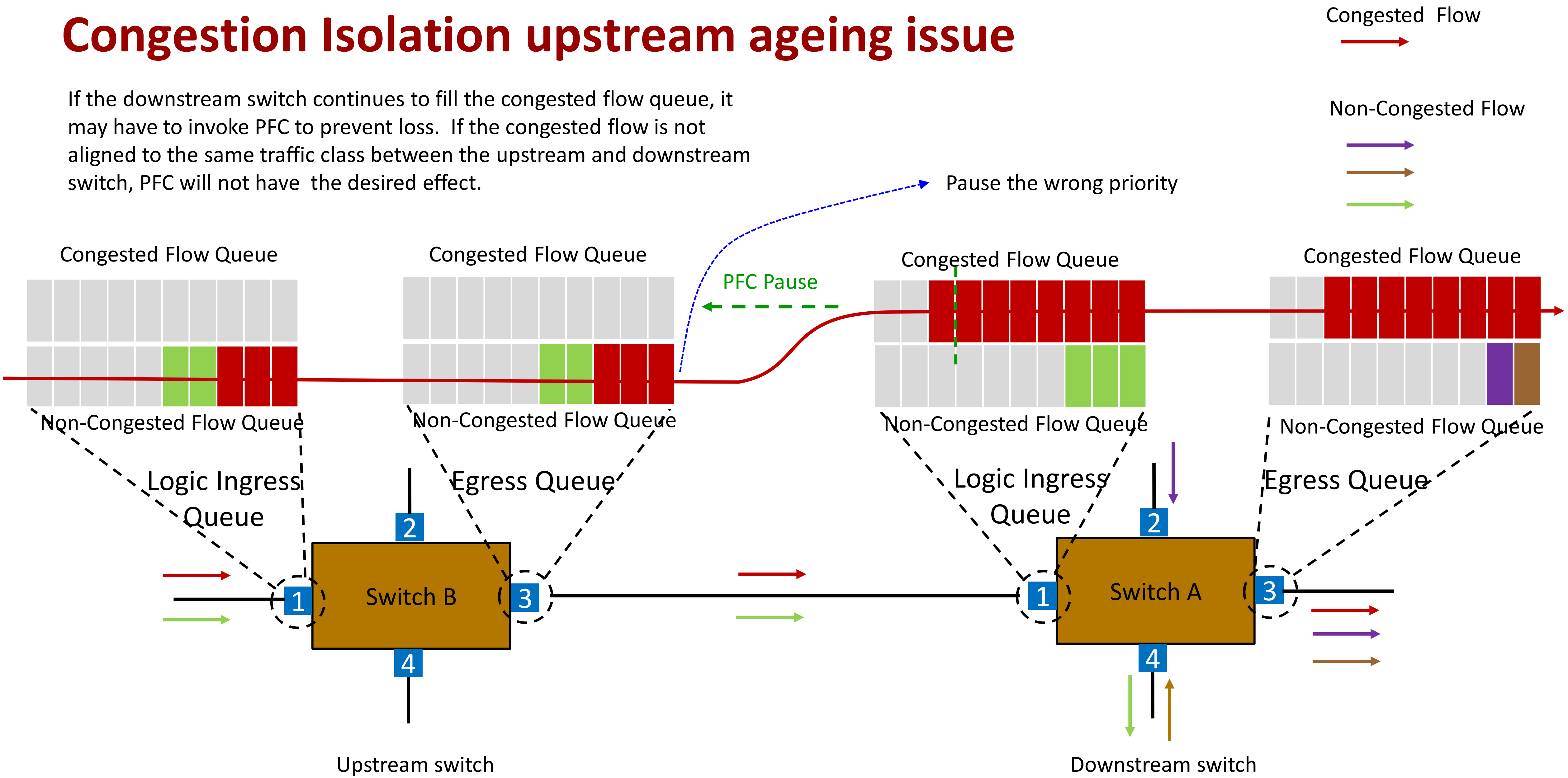
Congestion Isolation upstream ageing issue

If congested flows are allowed to return to the non-congested flow queue after some time period, it could be possible that the upstream switch does so asynchronous to the downstream switch.



Congestion Isolation upstream ageing issue

If the downstream switch continues to fill the congested flow queue, it may have to invoke PFC to prevent loss. If the congested flow is not aligned to the same traffic class between the upstream and downstream switch, PFC will not have the desired effect.



For simplicity, assume there is no other congested flow, so the bytes buffered in the egress congested queue is equal to the logic ingress congested queue.

Upstream ageing proposed resolution

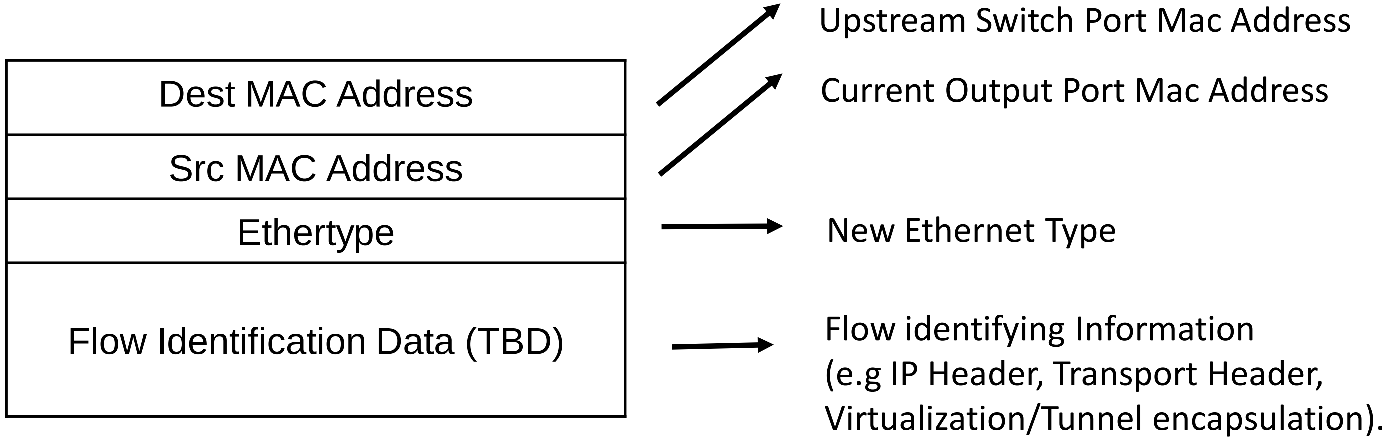
- Design space:
  1. Require upstream switch to use a longer ageing timeout (add epsilon to neighbor timeout)
  2. Never allow a flow to return to non-congested status
  3. Upstream switch can mark packets with egress traffic class
  4. Upstream switch can explicitly signal downstream when flow is moved to non-congested status
  5. Continue to signal upstream CIPs if congestion persists in congested queue
  6. Downstream switch signals another type of CIP when the flow is no longer congested
- Prefer Option #1
  - Neighbors exchange age timeout for congested flows in LLDP discovery TLV
  - When a flow entry is created due to the reception of CIP message from downstream neighbor, the ageing timeout should be larger than downstream neighbors age timeout.

Congestion Isolation Packet

- Objectives/Requirements:
  - Provide upstream neighbor with an indication that a flow has been isolated
  - Provide upstream neighbor with flow identification information
  - No adverse effects of single packet loss
  - Low overhead

- **NOTE:** Consider re-using 802.1Qau CNM format, but use upstream switch as DA MAC?

Format of Congestion Isolation Packet



Dest MAC Address

Src MAC Address

Ethertype

Flow Identification Data (TBD)

Upstream Switch Port Mac Address

Current Output Port Mac Address

New Ethernet Type

Flow identifying Information
(e.g IP Header, Transport Header,
Virtualization/Tunnel encapsulation).

Leverage existing CNM message?

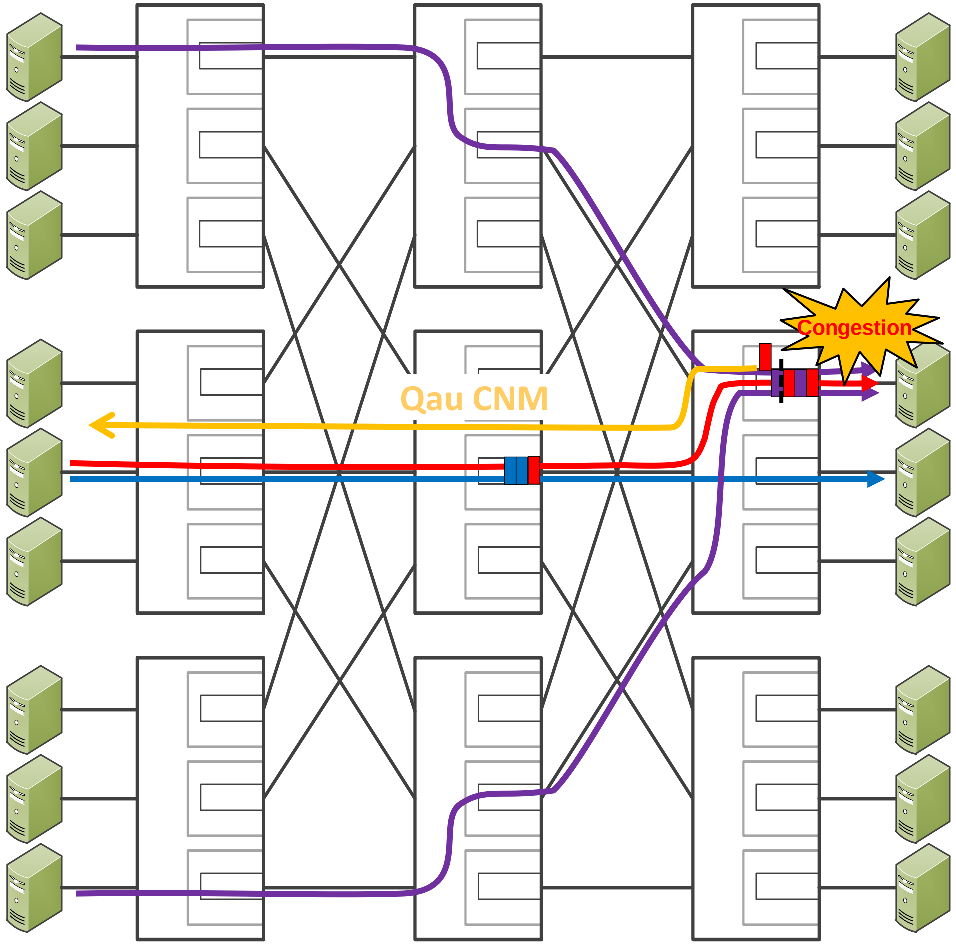
Congested Flow

Victim Flow

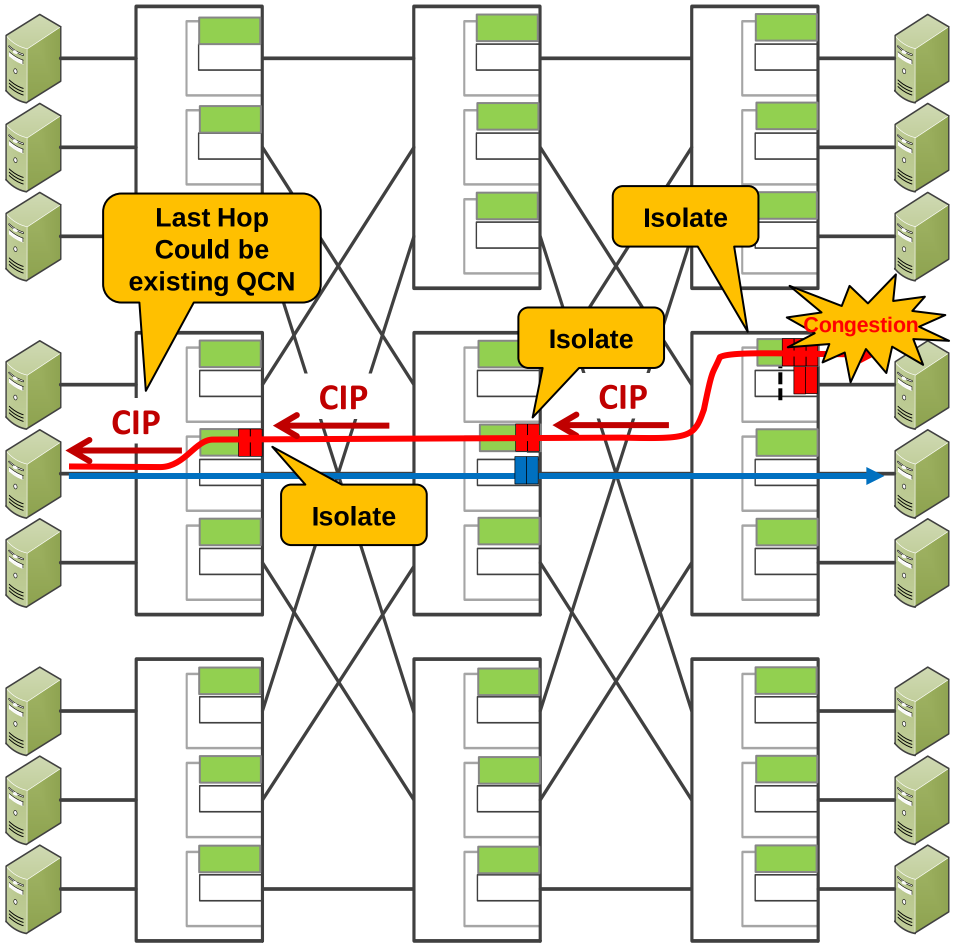
Congested Queue 🟩

Normal Queue □

End-to-End
QCN Congestion Notification Message



Hop-by-Hop
Congestion Isolation



Design Team discussion on CIP Format and CNM response

In consideration of using the existing Qau CNM message format for CIP, we have discussed the following

- Pros
  - Possible implementation leverage in switches if they implemented Qau
  - Possible that last hop to Server could simply use Qau “as is” if they implemented it.
- Cons
  - 64B might not be enough for IPv6 extensions and for last hop virtualization, they need to have the inner headers if encapsulated using VxLAN - so current format is likely not sufficient
  - Complications with IP options and non-standard headers.
- Proposal
  - Use existing format, but allow extension of additional bytes if requested by upstream neighbor
  - Consider additional LLDP TLV attributes to request CIP extension
  - Don't worry about the last hop to the NIC and consider not using CI in the last hop. We want ECN feedback to cause rate injection reduction, not CIP.

Order Preservation

- Agree that details are implementation specific. Externally observable behavior must be specified.
- At a minimum, uncongested to congested transition preserves order by strict priority service to the uncongested queue
- Congested to uncongested transition is done on:
  - Flow termination (namely, no such transition)
  - Inactivity timeout
  - Flowlet boundary (detected by inactivity timeout)
- Recommendation
  - An implementation MUST not increase the probability of out-of-order packets and can do so by using strict priority. An implementation SHOULD use other scheduling algorithms to avoid starvation, but this will be implementation specific and not otherwise specified.

Capability Discover via LLDP

- Objectives/Requirements:
  - Peer switches must know that each is capability of Congestion Isolation
  - Switches should agree on the traffic class used for the Congested Flow Queue
  - Switches should agree on the traffic classes that will monitored for congestion
  - Helpful to inform the upstream switch of the inactivity timeout used downstream so it may use a larger timeout to avoid early ageing.

Format of LLDP TLV

TLV Type	TLV info length	802.1 OUI	subtype	Congested Queue	Monitored Queues	Inactivity Timeout

Operation in hierarchical networks (e.g. VXLAN)

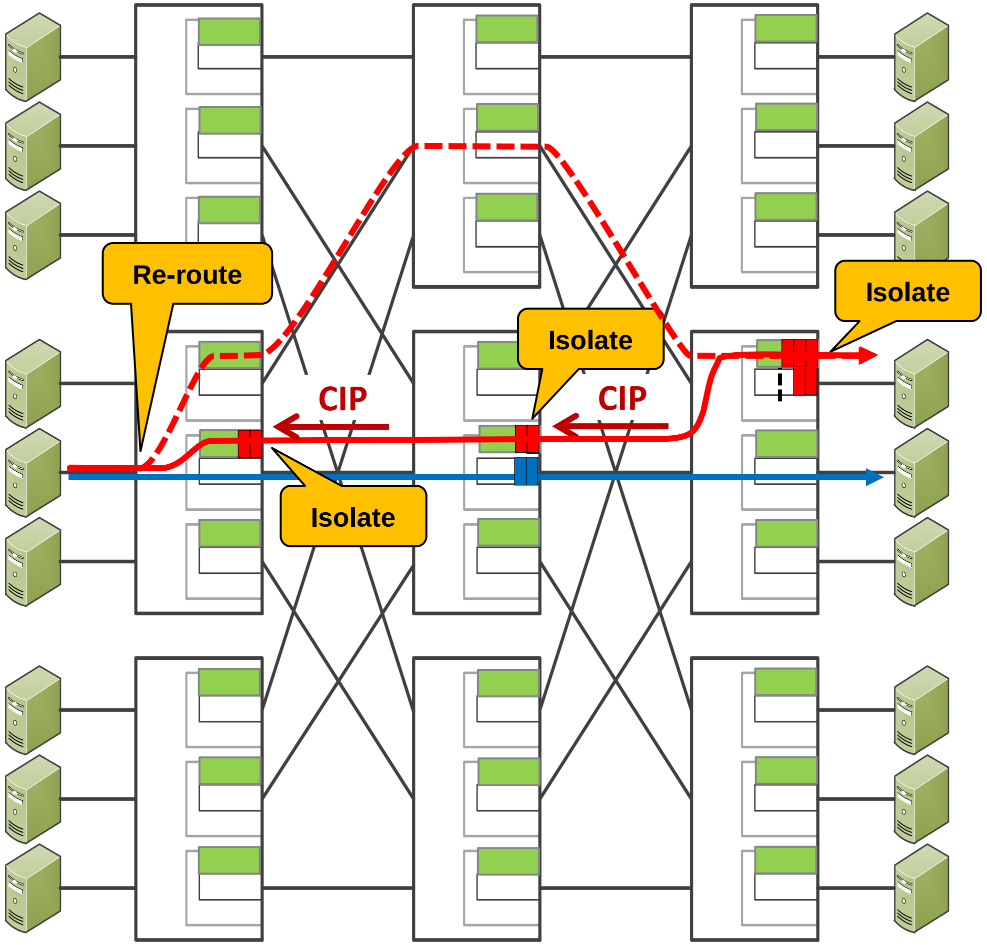
- In the core, each tunnel may include an aggregate of mice and elephants, are we after passenger flow tracking?
    - No, we only need to consider the outer tunnel header.
    - ECMP support causes load-balancing entropy to be put into UDP/TCP source ports, so flows are identified by outer tunnel header
- Any issues with operation on the overlay edge?
    - Assumption is that the Edge switch stores flows in their passenger format
    - Received CIP with <Encasulated SDU> may need to include passenger data
    - Challenges
        - Edge switch has to parse the flow from the CIP
        - Potentially <Encap SDU> has to be > 64B

Multicast Operation

- Does CI need to work differently for multicast traffic?
  - No CIP on Multicast?
  - CIP on Multicast + HoL blocking?
  - Configurable option (support multicast or not)?
- Discussion
  - Not much use of multi-cast in the DC, but CI should work the same.
  - Unclear how end-to-end congestion control works for multicast
    - All receivers can’t send feedback to the sender (scale issue)
  - Most multi-cast flows are throughput sensitive and not latency sensitive
  - Best to avoid configuration options if possible.
- Proposal
  - Since there isn't much use in the DC and it should work anyway, we propose to do nothing special about multicast in the CI standard.

Congested flows changing paths

Congested Flow →	Congested Queue 🟩
Victim Flow →	Normal Queue □



- An existing congested flow may arrive on a different source port
  - Network fault re-routing
  - Flowlet load-balance re-routing
- If the flow is still congested, egress ToR will need to send CIP to new upstream neighbor.
- Questions:
  - How to avoid CIP storm?
  - How to defend against miss-behaving ingress ToR that is spraying packets?

Congested flows changing paths

- There will need to be a state machine description of when to generate CIP messages.
  - NOTE: Could leverage Qau triggering mechanism and description
- There will need to be a specification of a flow table or a ‘stream identification function’
- This table will indicate if a flow is isolated or not and if a CIP has been sent, perhaps how many have been sent to support sending multiple CIPs to recover from loss.
  - NOTE: Qau triggering mechanism covers loss and multiple CIP generation
- The table could also register the source port a flow was last seen. If the source port is different, we could generate another CIP.
- May need to have some kind of rate control or counter that will limit the number of CIPs sent.

Recovery From Loss Of CIP Messages

- Problem statement: loss of CIP may result in packet loss
  - Downstream switch moves a flow to the congested queue → CIP is lost → Upstream switch keeps classifying the flow to the uncongested queue → Downstream switch issues PFC on the congested priority → Upstream switch doesn’t pause the flow
- Some Options:
  - Send multiple (such as 3) CIPs in succession to provide redundancy.
  - Send CIPs periodically. Upstream send an ACK when it has received a CIP. Downstream stops after receiving ACK.
  - Send CIPs periodically. Upstream mark the subsequent packets when it has received a CIP.
  - Use Qau algorithm for generating CIP messages (30.2.1) with increasing CIP messages as congestion worsens
- Resolution: the congested queue should be capable of triggering CIP messages
  - 802.1Qau defines the logic to control the rate of CIP, but doesn’t specify its granularity: per bridge or per congestion point (i.e. per egress queue)

Summary

- A number of detailed technical topics have been discussed and investigated
- No major stumbling blocks
- Appears to be some framework leverage from 802.1Qau that may focus/reduce the scope of effort on a project
- Continue discussion and investigate additional design topics

Thank you

www.huawei.com