

Cisco Catalyst 9000 Switching QoS with Silicon One ASICs

cisco Live !

Deep dive

Ninad Diwakar
Technical Marketing Engineer

Cisco Webex App

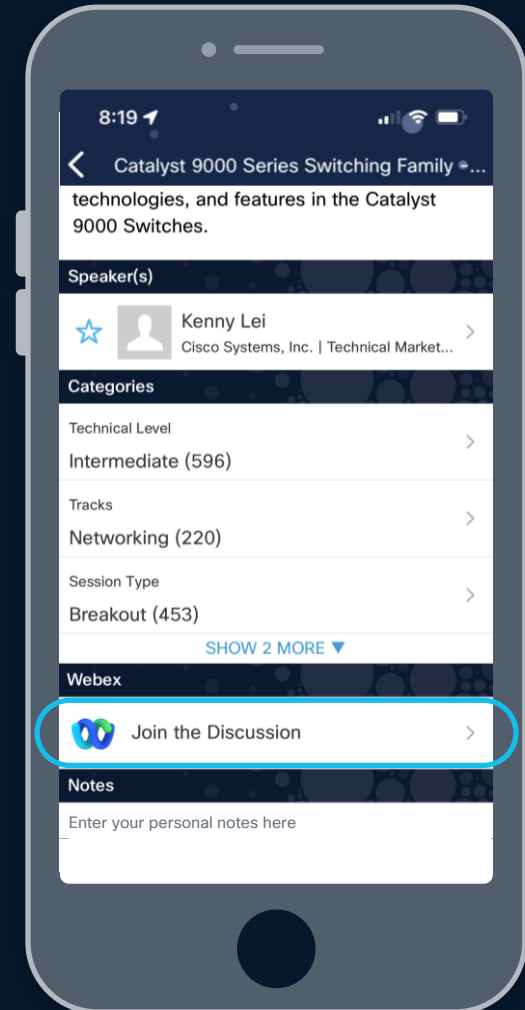
Questions?

Use Cisco Webex App to chat with the speaker after the session

How

- 1 Find this session in the Cisco Live Mobile App
- 2 Click “Join the Discussion”
- 3 Install the Webex App or go directly to the Webex space
- 4 Enter messages/questions in the Webex space

Webex spaces will be moderated by the speaker until June 13, 2025.



Agenda

01 Introduction to QoS

02 Silicon One QoS

03 Buffers

04 Config migrations

05 Closing

Introduction to QoS

Let's start with the fundamentals

What is Quality of Service?

- The primary **role of QoS** in Campus networks is to **manage packet loss**
- **Rich media applications** (audio/video) are extremely **sensitive to packet drops**
- In Campus networks – it takes only a few *milliseconds* of congestion to cause drops

QoS helps to differentiate and prioritize/de-prioritize traffic

Why we need QoS?

QoS as a feature – how do we classify

Critical features needed for day-to-day operations

OSPF

MPLS

EIGRP

Spanning tree

VLAN

MACSec/IPSec

Corner case/high availability features

SVL

Redundant Hardware

StackWise

Multi-chassis etherchannels

HSRP/VRRP

ESI mLag

QoS?

Always think of QoS in terms of worst-case scenario

Why we need QoS?

Qos defines the worst-case scenario

QoS configs define the worst-case scenario

You are not defining the best-case scenario

```
class USER-TRAFFIC
  bandwidth remaining percent 65
  random-detect dscp-based
  random-detect dscp af33 40 75
  random-detect dscp af32 60 90
  random-detect dscp af31 80 100
```

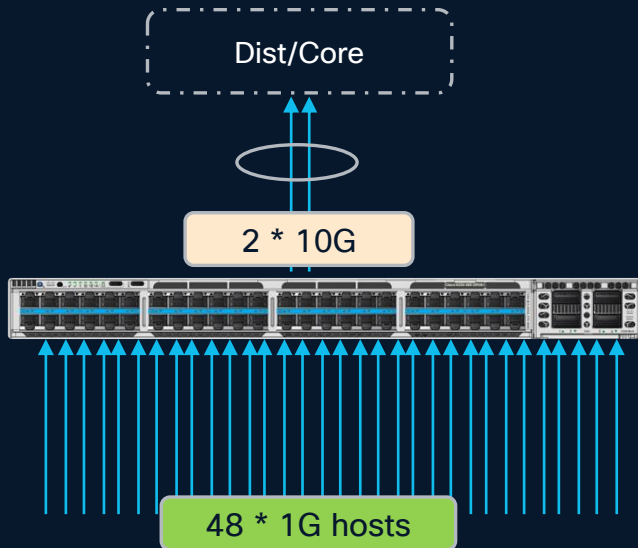
User traffic can take up to 65%* of bandwidth during congestion – You configure the Worst-case scenario

Best-case scenario is implicit.

Why we need QoS?

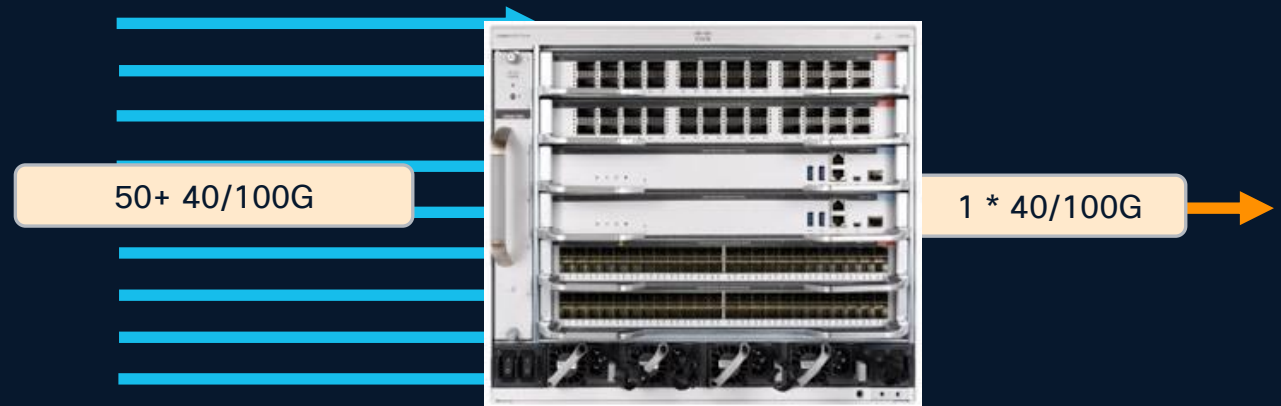
Congestion is more common than you think

At access level



Multiple hosts sending at close to line rate

At core/dist level



Single port going out to WAN/DC/Cloud/Remote branch

What does QoS do?

If we boil QoS down to a single concept, then..

QoS allows to split traffic on an interface into up to 8 different queues

Why should I care?

Traffic in each 'queue' can be treated differently

QoS helps to differentiate traffic and prioritize/de-prioritize traffic

Why do we need QoS

QoS helps to differentiate traffic and prioritize/de-prioritize traffic

Identify incoming traffic

Classification

Treat traffic differently

Scheduling and Queueing

Label traffic

Trust, Marking

Limit traffic

Policing and Shaping

What happens when queue is congested?

Congestion management (WTD and WRED)

How QoS works

Classification

Identify incoming traffic

Classification



Inspect and identify the traffic coming in (or going out)

Use logical rules

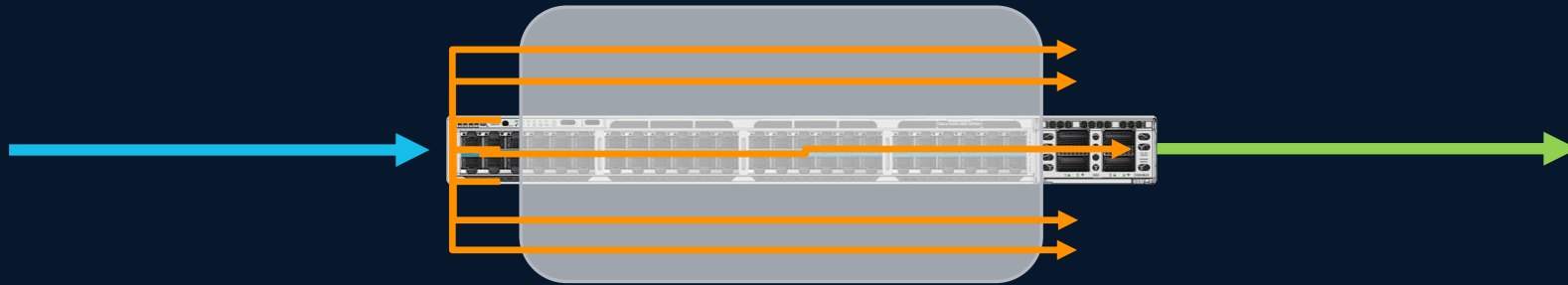
- Traffic per VLAN
- Match ACL
- Match packet tags – dscp or IP Prec or traffic-class
- Match protocol/traffic type

How QoS works

Queueing

Segregate the traffic

Queueing



After identification, separate to different queues. Max of 8 queues

Example:

- Traffic from one VLAN to a different queue.
- Traffic with a specific dscp tag to a separate queue.
- Specific protocol (example YouTube) to a separate queue.

How QoS works

Scheduling

Treat traffic differently

Scheduling



Logic to treat queues differently

Example:

- Traffic from one VLAN must never be dropped.
- Traffic with a specific dscp tag can be dropped second in the event of congestion.
- Specific protocol (example YouTube) can be dropped first in the event of congestion.

How QoS works

Marking

Label traffic

Trust, Marking



Rewrite (or retain) packet header (dscp) tags

Example:

- 'Label' the traffic coming from a specific host (ACL) with dscp 24.
- Devices upstream can now simply match 24 instead of classifying the acl again.

How QoS works

Congestion Management

What happens when queue is congested?

Congestion management (WTD and WRED)



Differentiate traffic within same queue.

Example:

- YouTube and discord go to the same queue.
- I want to drop YouTube traffic first and then discord traffic.

Let us start at the beginning

Queue types

Priority Queue

Traffic is very, very important – Think of an ambulance on a congested road.

Normal Queue

All other traffic.

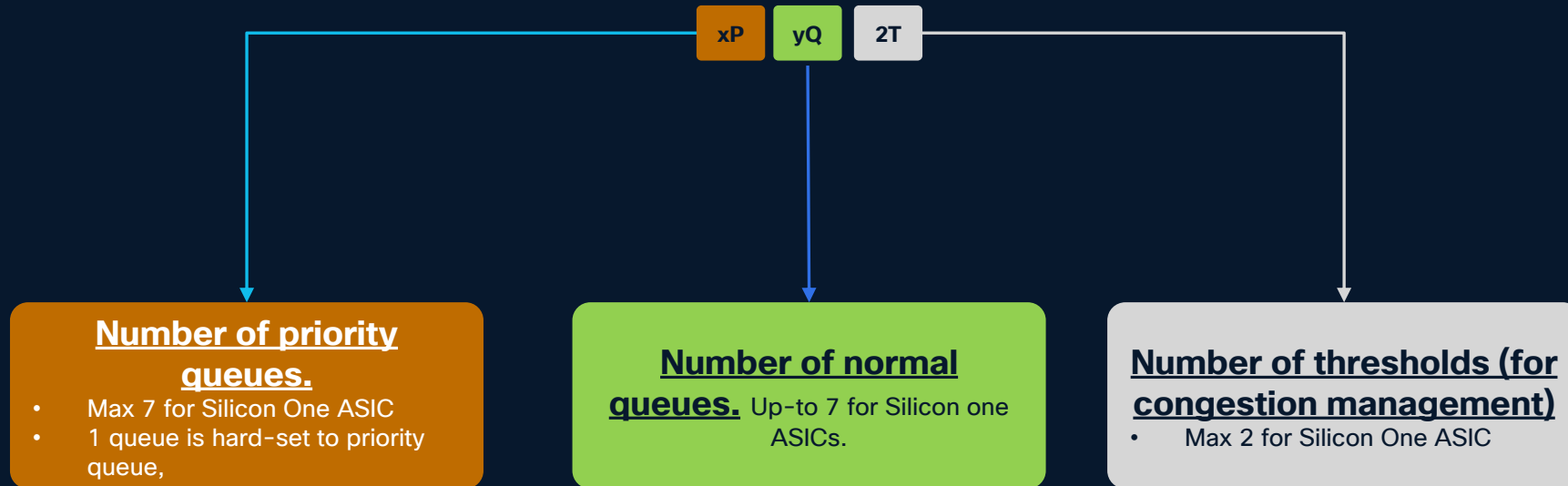
Why can't all traffic be priority?

If everything is priority, nothing is prioritized.

Think of an ambulance, with **100 ambulances on the road, normal traffic will come to a standstill**

Let us start at the beginning

Naming conventions



Priority Queue

Traffic given highest possible priority with the goal of sending this traffic out with minimum latency.

Normal Queue

Traffic is processed only if there is no traffic in priority queue.

Example

2P6Q2T

2 **P**riority queues, 6 normal **Q**ueues and 2 **T**hresholds

Catalyst 9000

Switching family

UADP 2.0x

UADP 3.0x

Catalyst® 9200CX Series



Catalyst 9300X models



Catalyst 9200 Series



Catalyst 9300 Series



Catalyst 9400 Series



Catalyst 9500 Series



Catalyst 9600 Series with Sup-1



Catalyst 9400X models



Catalyst 9500 High performance Series



Class-based Weighted Fair Queue (CBWFQ) QoS

S1 Q200

NEW S1 A100

NEW S1 K100/E100

Catalyst 9600 Series with Sup-2





C9350 Series



C9610-SUP3/C9610-SUP3XL

Virtual Output Queue (VoQ) QoS

QoS Challenges

Syntax differences

Catalyst 6500

```
interface TenGigabitEthernet2/14
 platform qos queue-mode mode-dscp
 wrr-queue bandwidth 20 1 14 10 15 2 3
 priority-queue queue-limit 20
 wrr-queue dscp-map 1 1 0 1 2 3 4 5 6 7
 wrr-queue dscp-map 2 1 8 14 32 35 36 37 38
 wrr-queue dscp-map 3 1 10 22 24 30 49 50 51 52
 wrr-queue dscp-map 4 1 16 18
 wrr-queue dscp-map 5 1 26
 wrr-queue dscp-map 6 1 48 56
 wrr-queue dscp-map 7 1 34
```

Can I copy over old configuration?

Watch me reject every line of this config!

Catalyst 9300

The solution?

MQC CLI – Example on UADP

class-map

Identify the traffic we care about

```
class-map match-any Voice
  match dscp ef
class-map match-any Video
  match dscp 34
```

policy-map

Prioritize or limit traffic

```
Policy-map POLICY-QOS
  class Voice
    priority level 1
  class Video
    set dscp 10
```

service-policy

Apply on an interface.

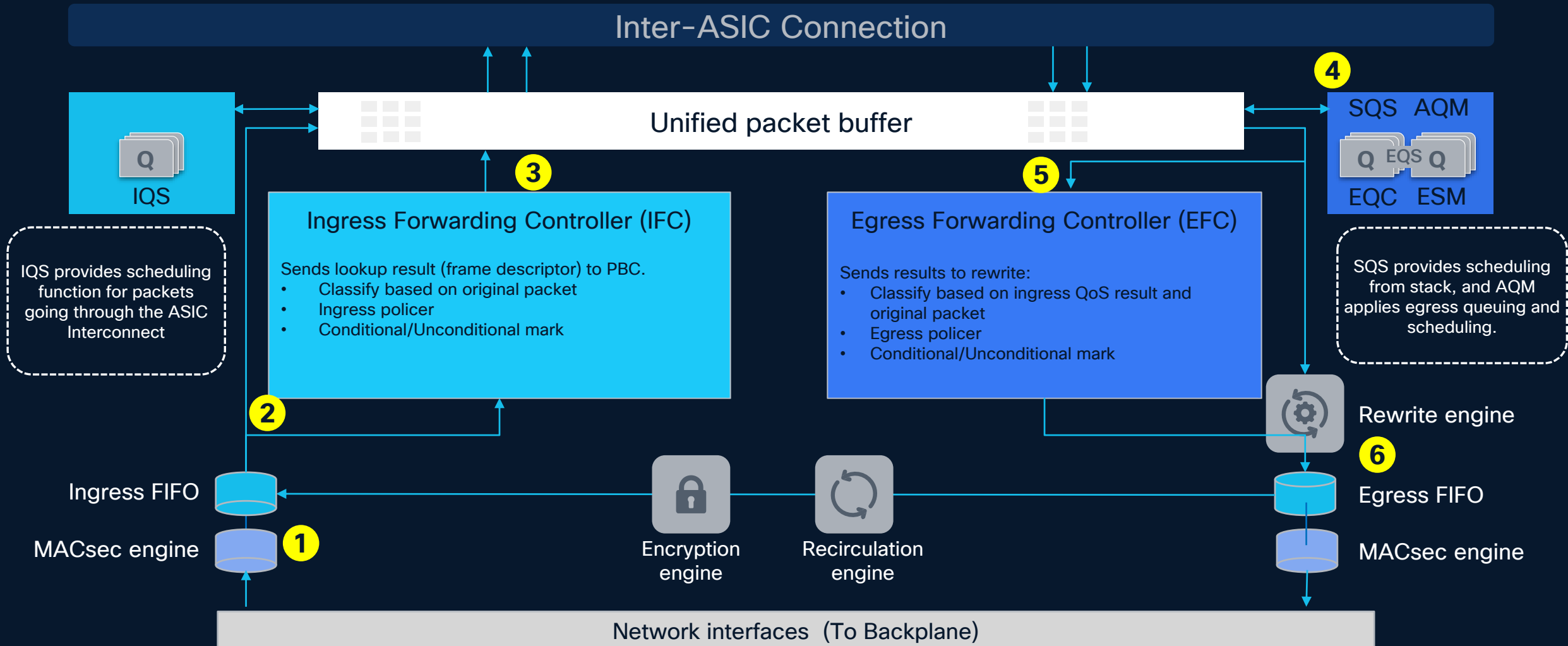
```
interface x/y
  service-policy (input/output) POLICY-QOS
```

This will still be rejected on Silicon One switches. But the config is readable

Silicon One QoS

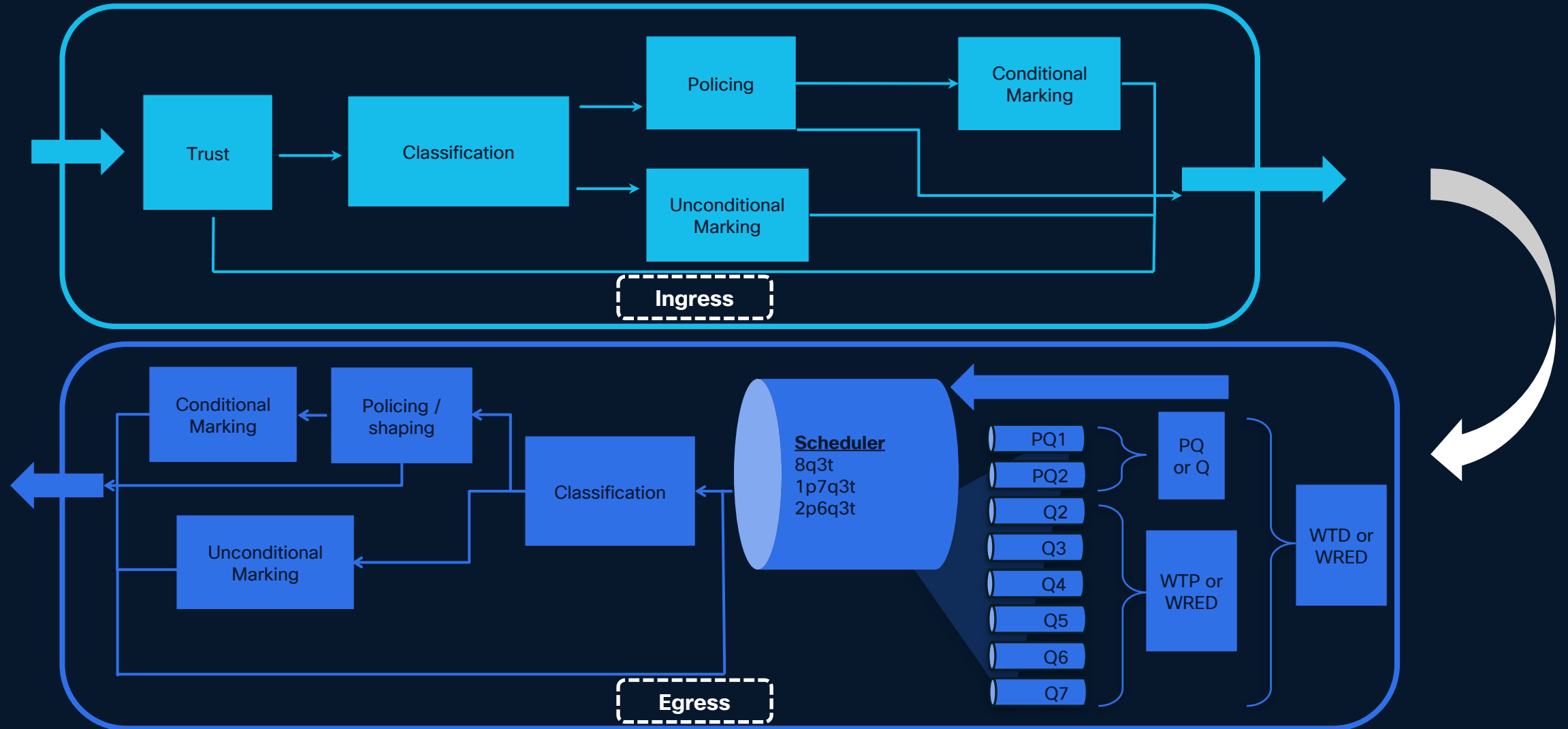
UADP QoS Recap

Packet walk



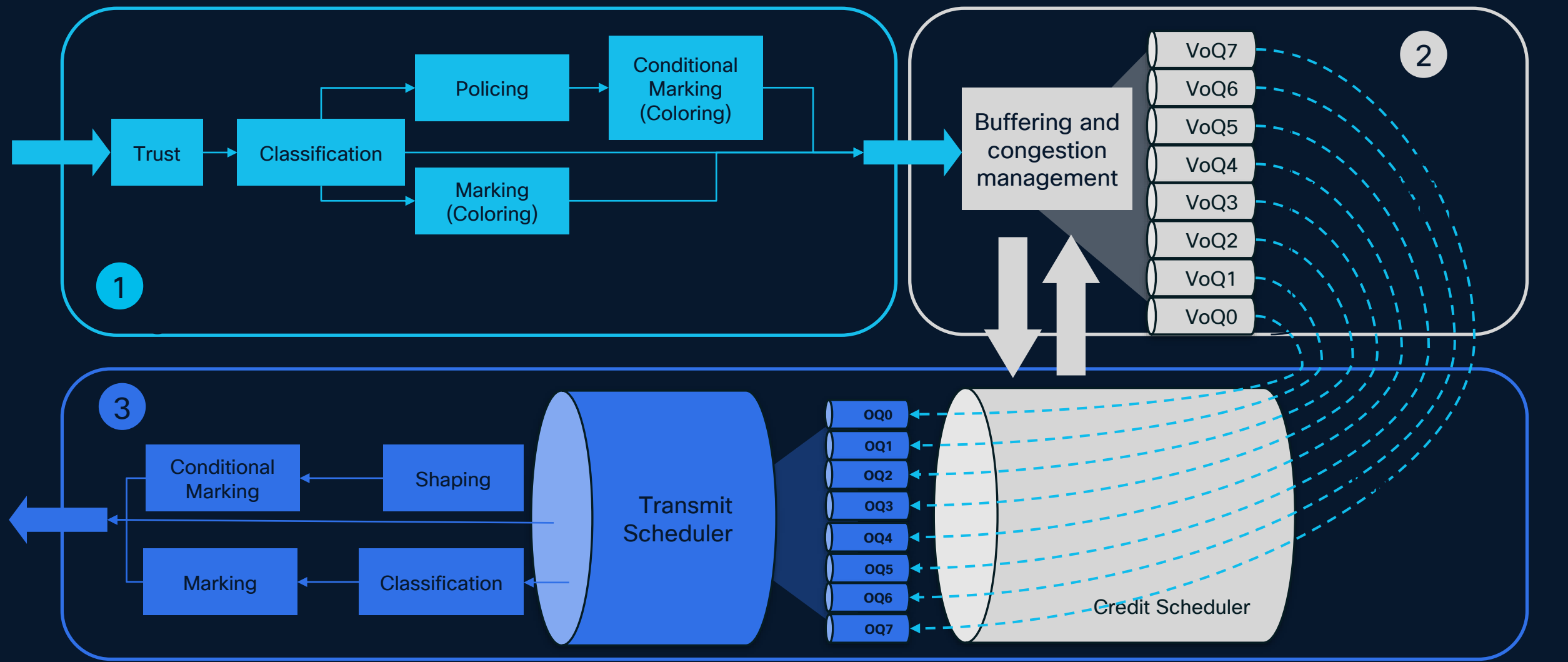
UADP QoS Recap

Actions in UADP



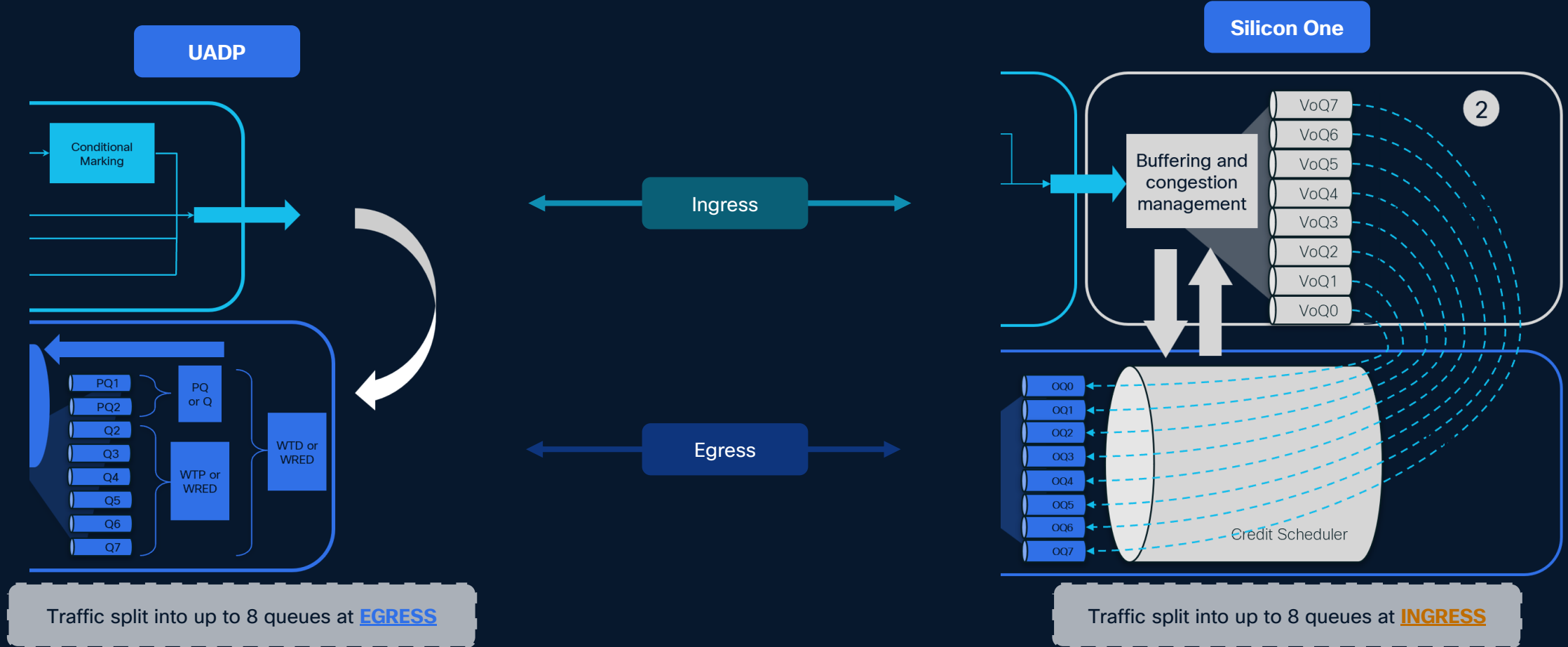
WRED: up to 4 queues with UADP 2.0x; up to 8 queues with UADP 3.0x

Silicon One



QoS architecture

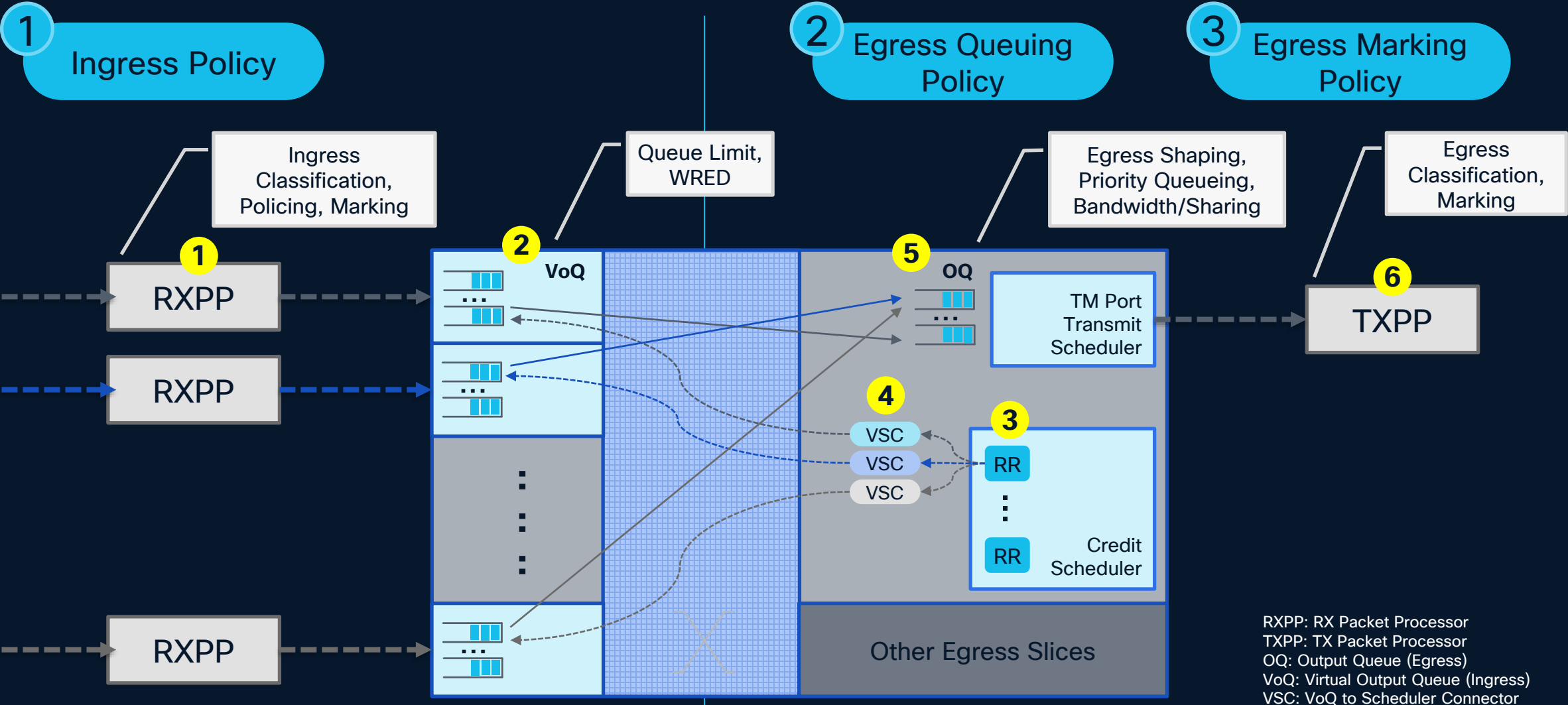
Silicon One vs UADP



Why should I care?

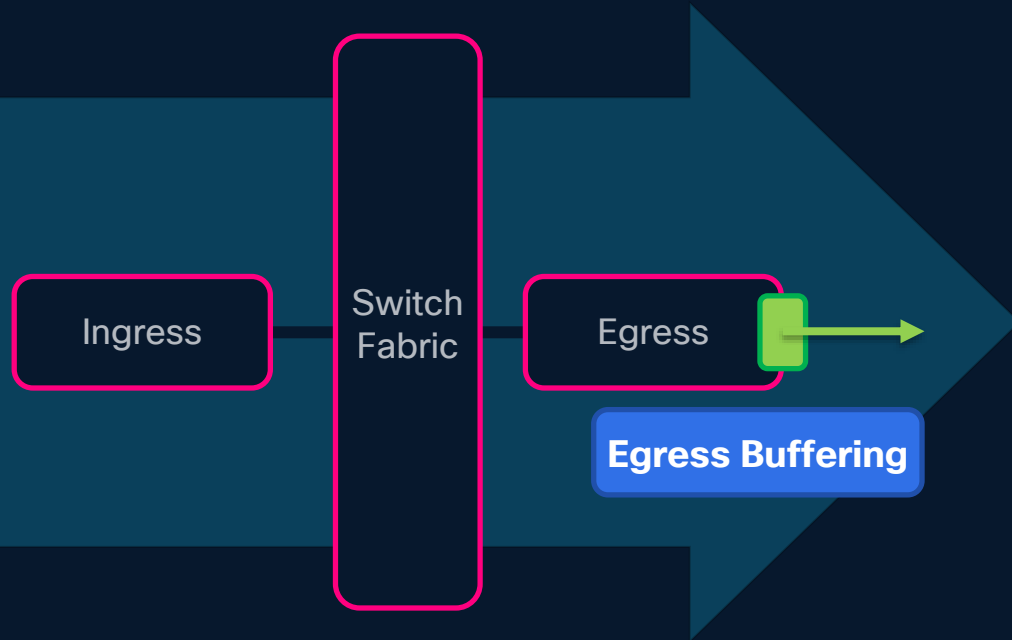
- Queue drops occur at incoming port **before the switch resources are used** for look-up/forwarding.
- Granular control over traffic egressing from one port with multiple incoming ports -> **many to one granularity**.

Silicon One ASIC mapping

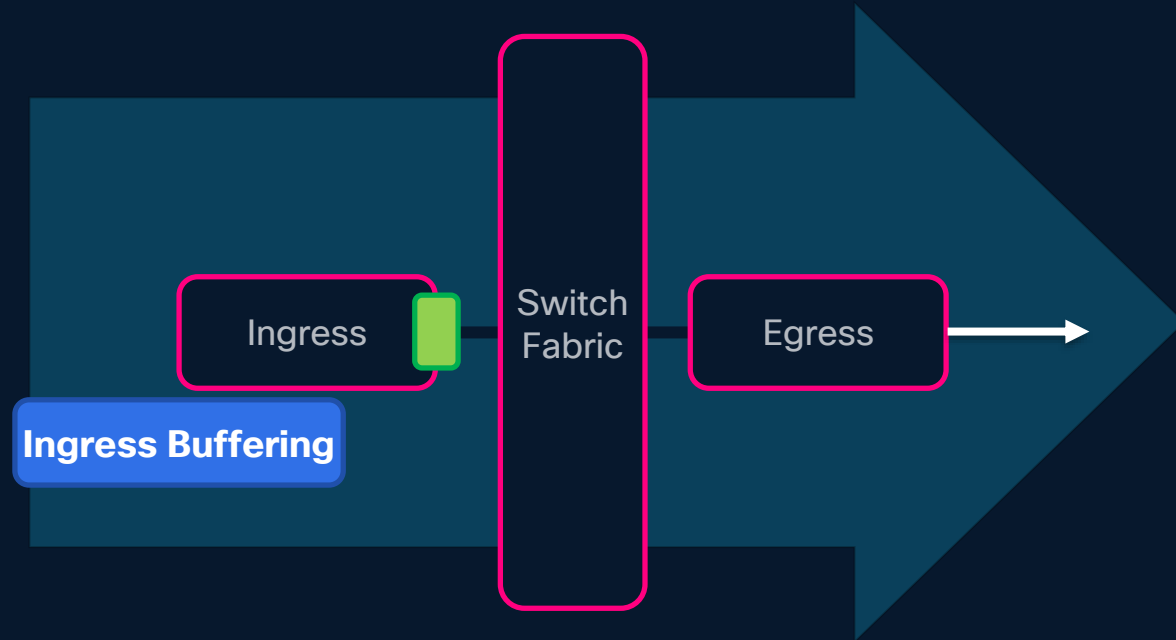


Buffer types

Silicon One vs UADP



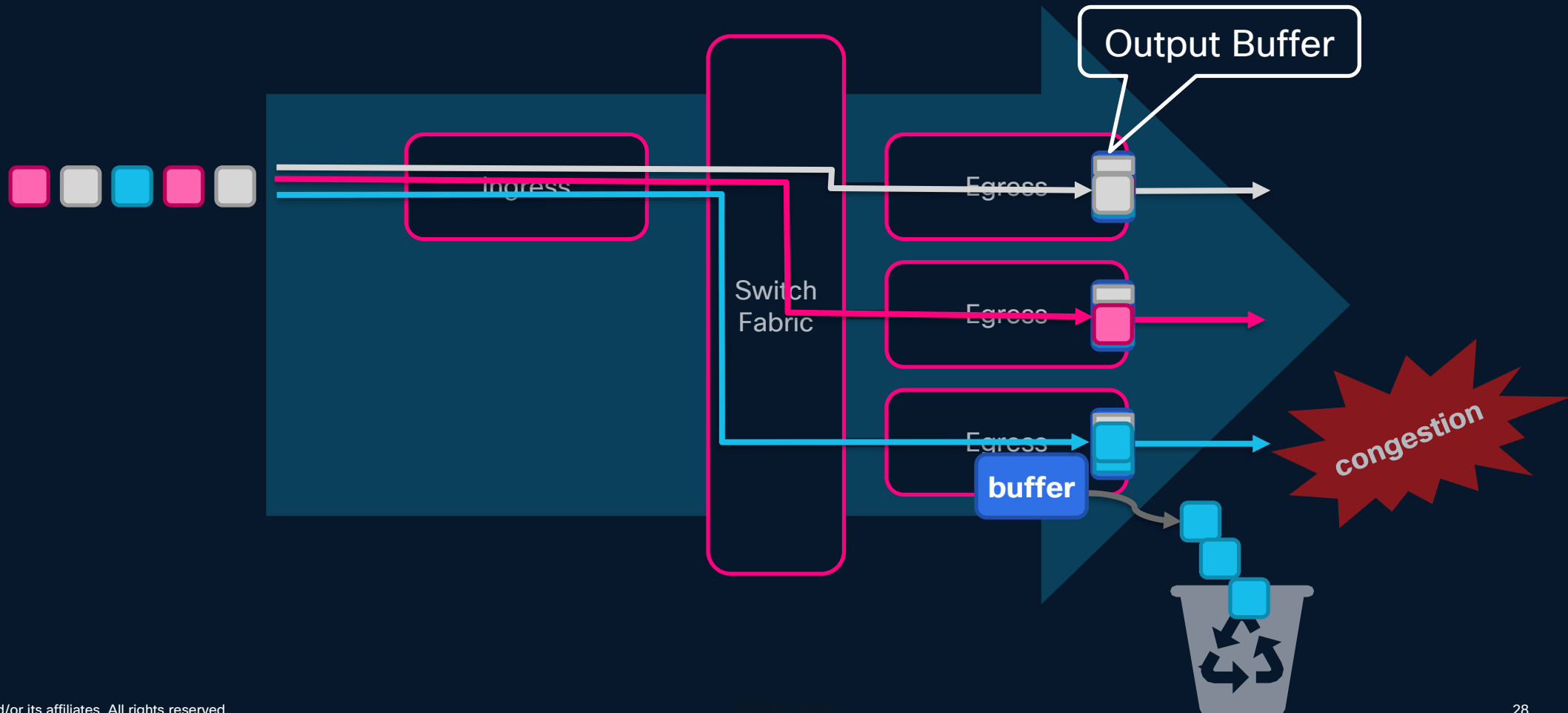
UADP



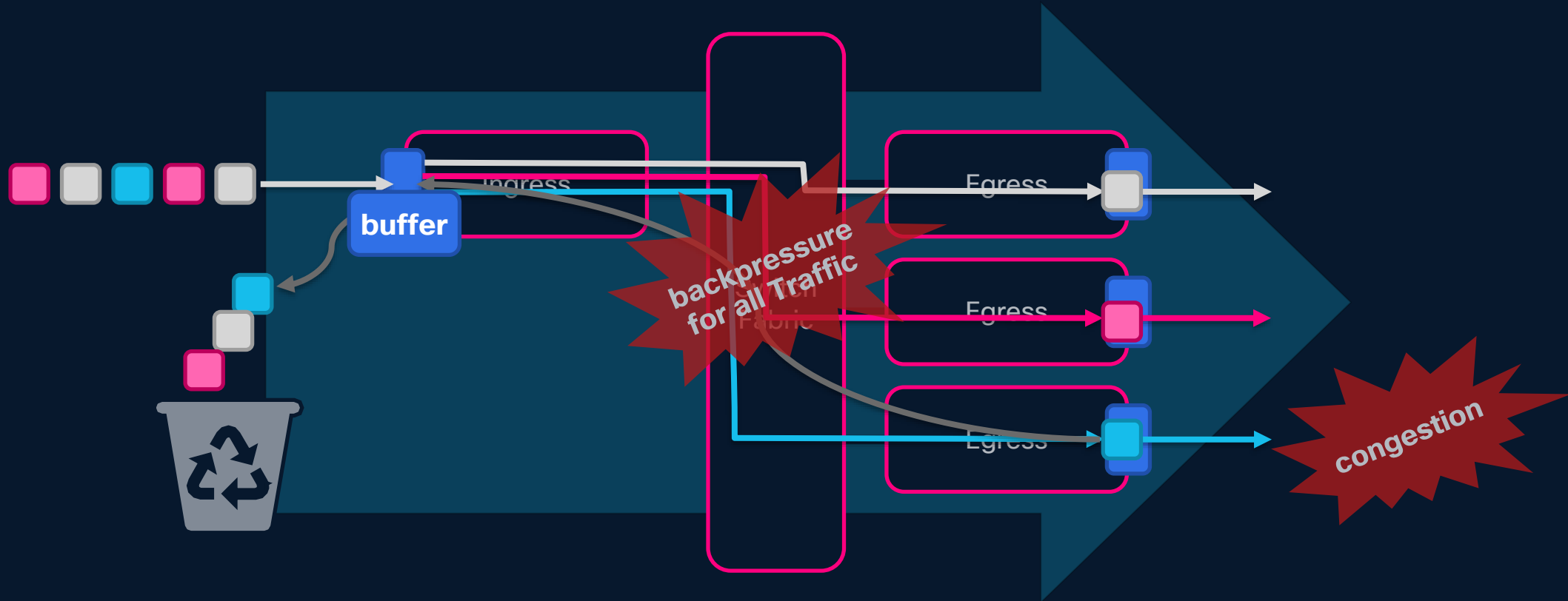
Silicon One
Q200

Egress buffering - UADP

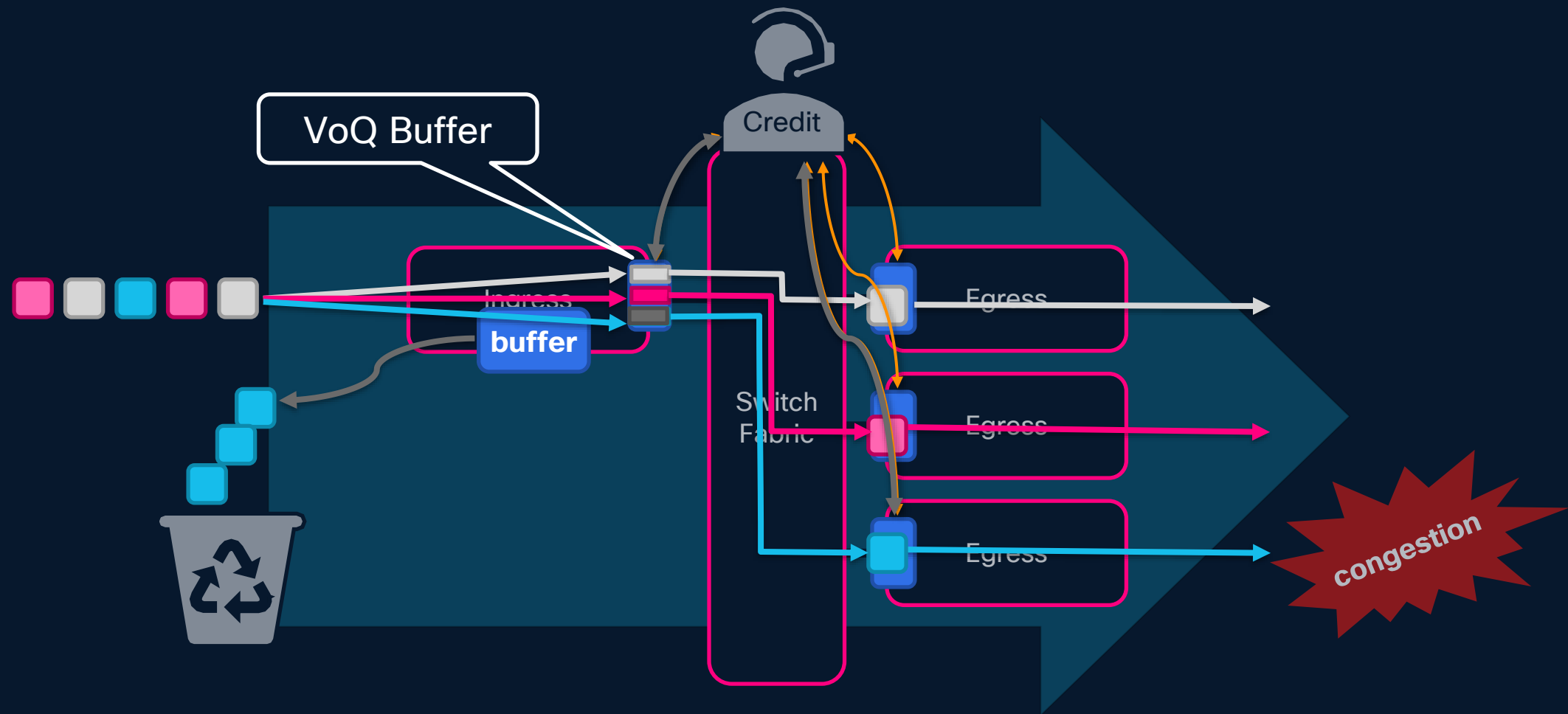
- No Head of Line Blocking



Ingress buffering – Head of Line Blocking



Virtual Output Queuing – Silicon One ASIC



QoS Fundamentals

dscp, cos, ip prec and traffic-class

UADP Devices



Can forward and define priority using [dscp, cos or ip prec](#)

Silicon One Devices



Can **ONLY** forward and define priority using [Traffic-class](#)

QoS Fundamentals

Silicon One ASIC – Traffic-class

VoQ devices



- Additional field appended to the packet header while processing within the switch.
- Field added when packet enters the switch
- Field removed when packet exits the switch
- Neighbouring switches have **no idea about traffic-class**

QoS Fundamentals

Silicon One ASIC – Traffic-class

UADP 2/3

Forwards and defines priority using dscp, cos or ip prec

**Silicon
One**

Forwards and defines priority using **Traffic-class**

This adds a mandatory overhead to configuration

QoS Fundamentals

What is the overhead?

INGRESS POLICY

Marking

Map traffic header/ACL/VLANs to traffic-class. Color traffic if congestion management is needed.

EGRESS POLICY

Queueing

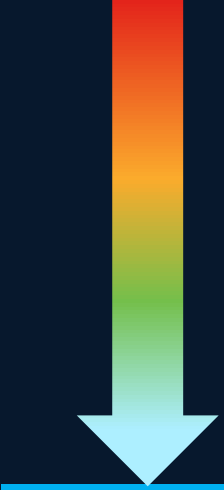
Perform queueing based on traffic-class. Perform WRED/WTB based on packet color.

Queueing without assigning traffic-class at ingress will **NOT work**

Map to traffic-class

But what is traffic-class?

- Additional tag appended to the packet on entry.
- 3-bit value.
- Can have 8 possible values (0-7)
- Tag removed when packet exits the switch

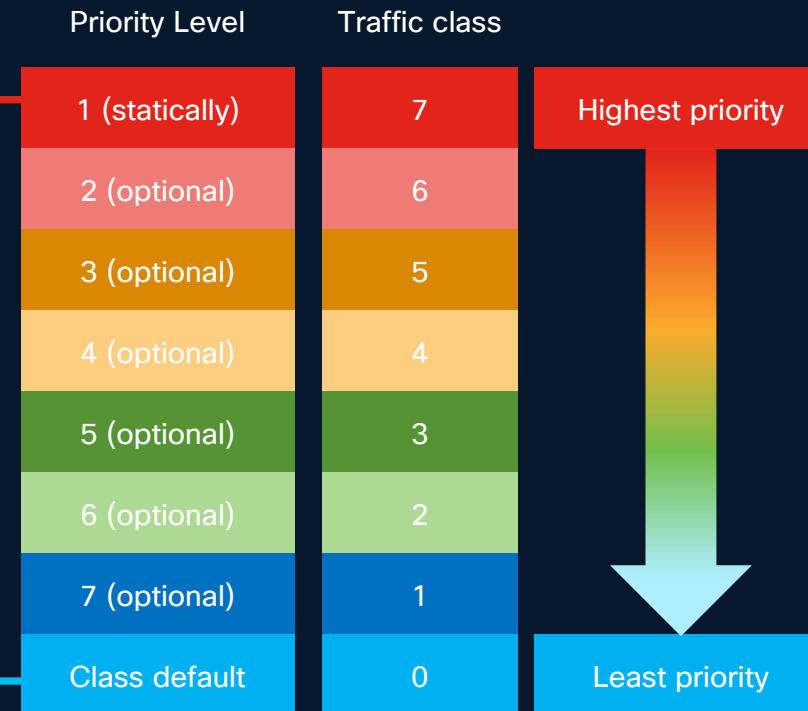
Priority Level	Traffic class	
1 (statically)	7	 Highest priority Least priority
2 (optional)	6	
3 (optional)	5	
4 (optional)	4	
5 (optional)	3	
6 (optional)	2	
7 (optional)	1	
Class default	0	

Traffic-class

Understanding priority

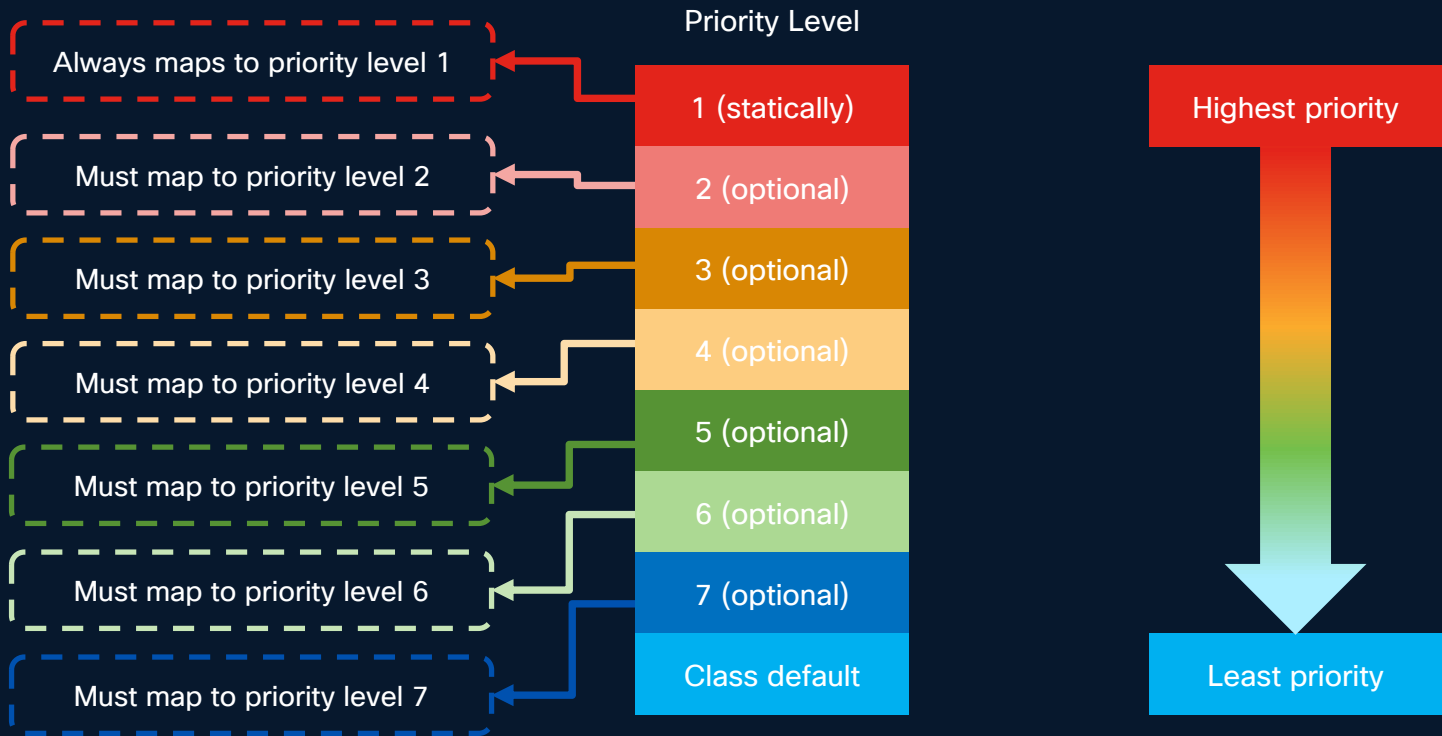
- Traffic-class 1 will always be a **strict priority queue at level 1**
- This **cannot** be changed via config.
- 8Q configuration is not supported. 1P7Q and above (2P6Q and so on) is supported.

- Any class not matched will automatically be sent to traffic-class 0.
- This class is always enabled.
- You cannot match other traffic-classes (example tc5, tc4) etc to class-default
- Class-default **cannot be a priority queue**. Must be normal queue (8P not supported)



Traffic-class

Understanding priority



Traffic-class

Let us look at some examples

```
policy-map type queueing EGRESS
class tc7
  priority level 2
  shape average percent 10
class tc6
  priority level 3
class tc5
  bandwidth remaining ratio 10
class class-default
  bandwidth reaming ratio 35
```



```
policy-map type queueing EGRESS
class tc6
  priority level 2
class tc5
  bandwidth remaining ratio 10
class class-default
  bandwidth reaming ratio 35
```



- Tc7 must always be priority level 1
- If tc6 is configured as priority queue – it must be priority level 2

- Priority must be enabled in sequence.
- Without priority level 1, we cannot enable priority level 2.

```
policy-map type queueing EGRESS
class tc7
  priority level 1
  shape average percent 10
class tc5
  bandwidth remaining ratio 10
class class-default
  bandwidth reaming ratio 35
```



```
policy-map type queueing EGRESS
class tc7
  priority level 1
class tc6
  priority level 2
class tc5
  priority level 3
class tc4
  priority level 4
class class-default
  bandwidth reaming ratio 35
```



- Tc7 is priority level 1
- We do not have tc6, but since we have tc5 as a normal queue, this is supported.

- This is an example of 4P1Q. Since priority follows the traffic-class priority, this is supported.

Mapping traffic-class to VoQ

Your ingress policy-map creates the VoQs.

No ingress policy-map, we use the **default VoQs**

Silicon One

Default VoQ

Traffic class	
7	DSCP: 16, 24, 32, 46, 48, 56 COS: 4, 5
6	VoQ 6-1 are <u>disabled</u>
5	
4	
3	
2	
1	
0	DSCP: All other values; COS: All other values All EXP values

Only 2 VoQs (tc7 and tc0) are active

VoQ mapping

Let us look at an example

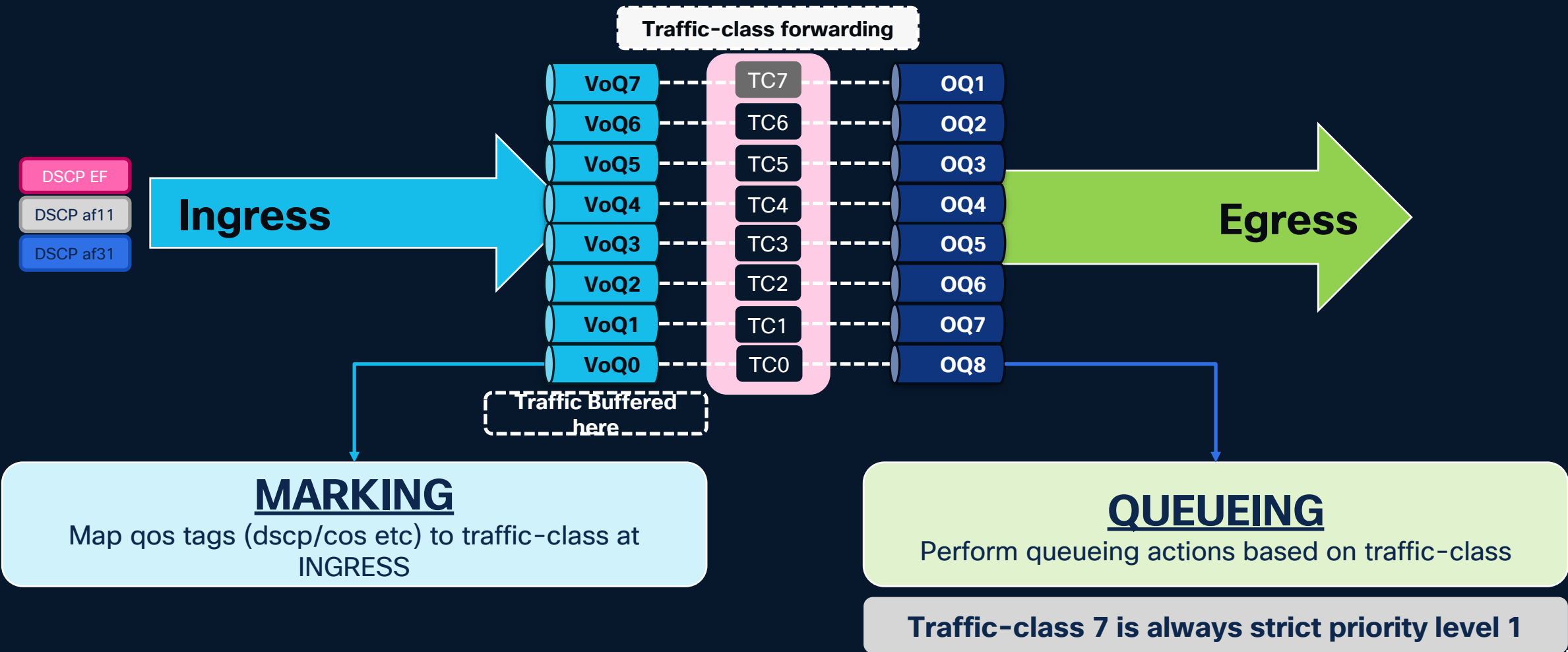


- Predefined tags (example 26, ef etc) go to tc7, and everything else goes to tc0.
- Tc6 and tc5 will **NOT be used** as those VoQs are disabled.

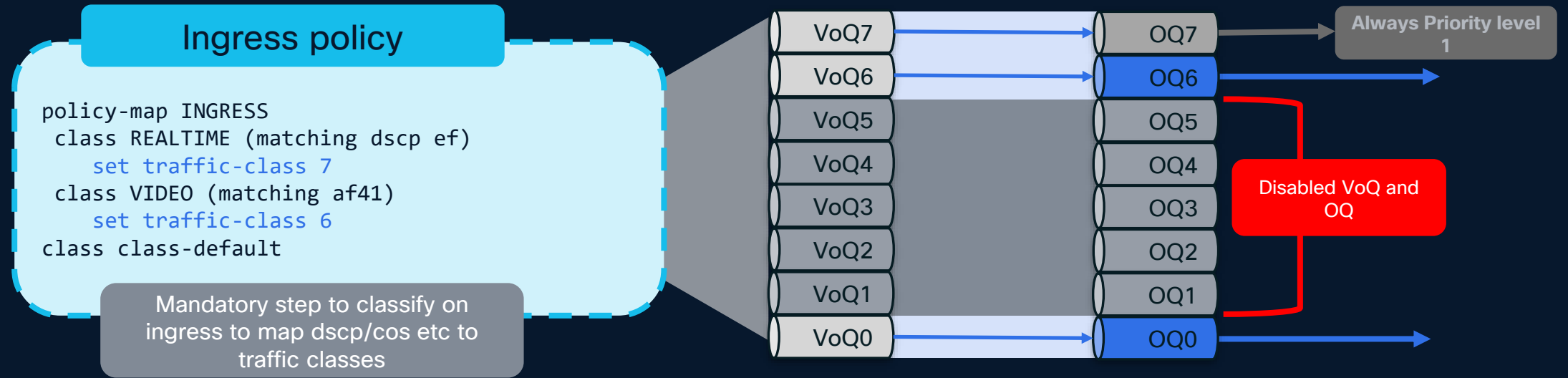
QoS Fundamentals

Activate VoQs at ingress

Map dscp to traffic-class and then use traffic-class at egress



Traffic class to VoQ mapping



- Ingress policy determines how many VoQ are enabled.
- Each VoQ has a 1:1 mapping with corresponding OQ.
- Traffic class 7 is always a strict priority queue regardless if configured or not.

VoQ mapping

Let us complicate it a bit

```
policy-map INGRESS-1
  class REALTIME (matching dscp ef)
    set traffic-class 7
  class VIDEO (matching af41)
    set traffic-class 6
  class class-default
```

```
policy-map INGRESS-2
  class REALTIME (matching dscp ef)
    set traffic-class 7
  class VIDEO (matching 26)
    set traffic-class 6
  class NETWORK (matching af41)
    set traffic-class 5
  class class-default
```



Egress policy

```
policy-map type queueing EGRESS
  class tc7
    priority level 1
    shape average percent 10
  class tc6
    priority level 2
  class tc5
    bandwidth remaining ratio 10
  class class-default
    bandwidth remaining ratio 35
```

AF41 from port 1 goes to traffic-class 6 to egress-port.
AF41 port 2 goes to traffic-class 5 to egress-port.

Silicon One QoS

Let's talk about naming

Ingress policy

```
policy-map INGRESS-1
  class REALTIME (matching dscp ef)
    set traffic-class 7
  class VIDEO (matching af41)
    set traffic-class 6
  class class-default
```

Ingress policy supports custom named classes

Egress policy

```
policy-map type queueing EGRESS
  class tc7
    priority level 1
    shape average percent 10
  class tc6
    priority level 2
  class tc5
    bandwidth remaining ratio 10
  class class-default
    bandwidth reaming ratio 35
```

Egress policy **does NOT support** custom named classes

Silicon One QoS

Classification and marking



Ingress classification

```
Class-map match-any Critical
  match dscp ef

Class-map match-any Servers
  match access-group name Server_IPv4

Class-map match-any IOT-Traffic
  match vlan 10

Class-map match-any Webex-traffic
  match protocol webex
```

Ingress marking

```
Policy-map ingress-marking
Class Critical
  set traffic-class 7
Class Servers
  set traffic-class 6
Class IOT-Traffic
  set traffic-class 5
Class Webex-traffic
  set traffic-class 4

Interface gi1/0/1
  service-policy input ingress-marking
```

Silicon One QoS.

Classification and marking contd



Egress classification

```
Class-map match-any tc7
  match traffic-class 7

Class-map match-any tc6
  match traffic-class 6

Class-map match-any tc5
  match traffic-class 5

Class-map match-any tc4
  match traffic-class 4
```

Egress queueing

```
Policy-map type queueing egress-queueing
Class tc7
  priority level 1
Class tc6
  priority level 2
Class tc5
  bandwidth percent 10
Class tc4
  bandwidth percent 15
Class class-default
  bandwidth percent 45

Interface gi1/0/3
  service-policy type queuing output egress-queueing
```

Silicon One QoS

Classification and marking contd.

Egress queueing

```
Policy-map type queueing egress-queueing
```

```
Class tc7
```

```
    priority level 1
```

```
Class tc6
```

```
    priority level 2
```

```
Class tc5
```

```
    bandwidth percent 10
```

```
Class tc4
```

```
    bandwidth percent 15
```

```
Class class-default
```

```
    bandwidth percent 45
```

```
Interface gi1/0/3
```

```
    service-policy type queuing output egress-queueing
```

I have created 2 priority queues (tc7 and tc6)

And 3 normal queues (tc5, tc4 and tc0)

This is a 2P3Q configuration

Silicon One QoS

What is a priority queue?

```
Policy-map type queueing egress-queueing
  Class tc7
    priority level 1
  Class tc6
    priority level 2
```

I have created 2 priority queues (tc7 and tc6) (assume a 8 queue config)

When I have a packet in the priority queue, STOP ALL traffic in other queues and send out the priority packet first.

Priority level 1 is higher than level 2 and higher than level 3 and so on.

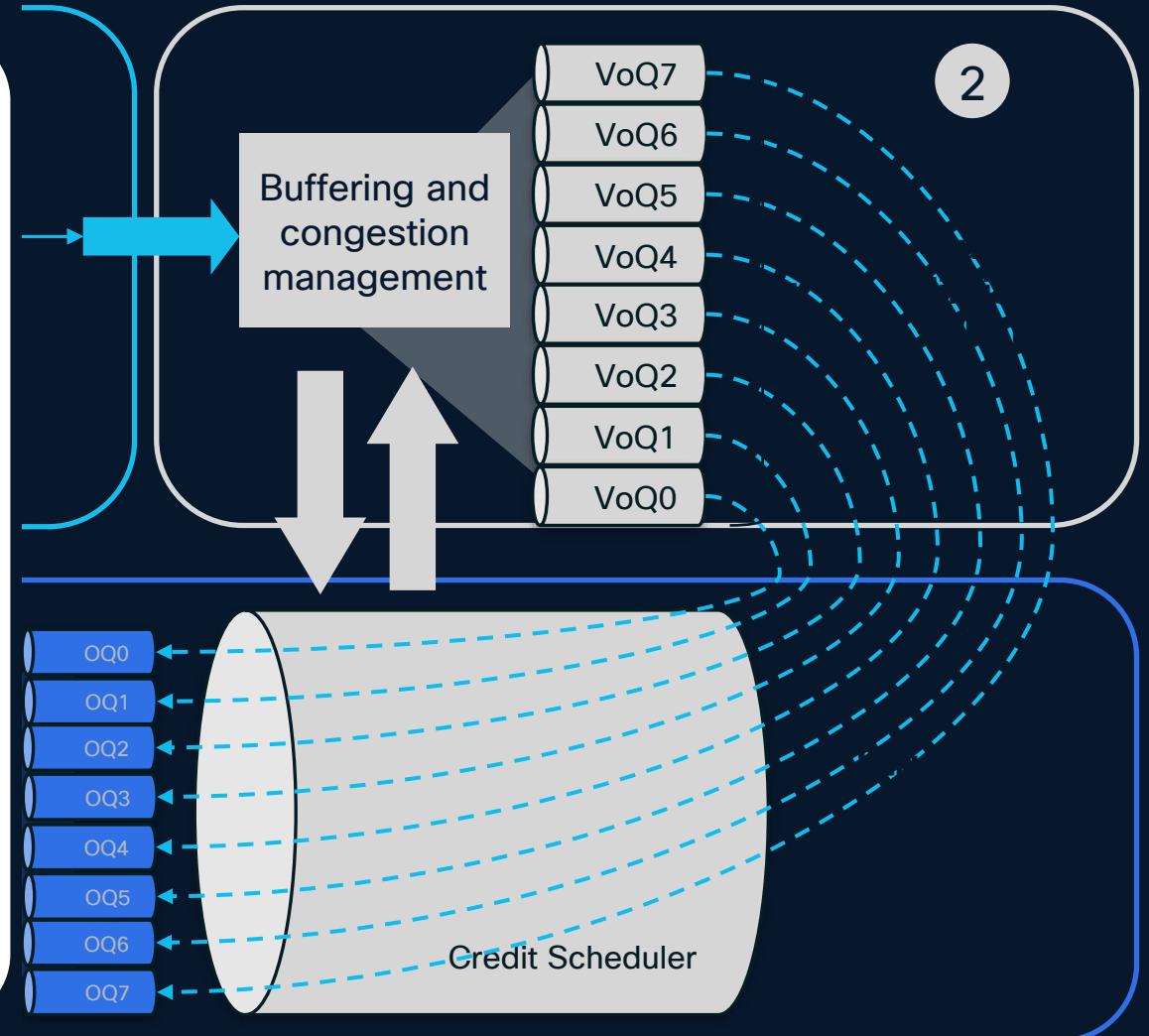
If packet comes on level 1 and 2, STOP level 2, send level 1 first.

Up to 7 priority queues (except tc0) can be configured. Priority level 1 to priority level 7.

Silicon One QoS

Let's talk scheduling

- Packet schedule from VoQ to OQ based on a credit scheduling system
- Packets are buffered at ingress (VoQ)
- Different type of queues are served differently
 - Strict priority queues
 - Always serviced first
 - Up to 7 PQs
 - Normal queues (without priority configured)
 - Served only after priority queues are empty
 - Use **Weighted Round Robin (WRR)** for scheduling



Silicon One QoS

Understanding weights

```
Policy-map type queueing egress-queueing
Class tc7
    priority level 1
Class tc6
Class tc5
Class tc4
Class class-default
```

1 priority queue and 4 normal queues (1P4Q)
Traffic in 4 normal queues are all treated equal
Tc6 is treated same as tc5, tc4, tc0

Use '**weights**' to prioritize some classes over others.

```
Policy-map type queueing egress-queueing
Class tc7
    priority level 1
Class tc6
    bandwidth percent 10
Class tc5
    bandwidth percent 50
Class tc4
    bandwidth percent 60
Class class-default
    bandwidth percent 10
```

For every 5 packets then tc5 sends, tc4 can send 6, and tc0 and class-default can send 1.

Silicon One QoS

Limiting traffic

Ingress policing

```
policy-map test-police-1R2C
class dscp1
  set traffic-class 3
  police rate 10g bps
  conform-action transmit
  exceed-action drop
```

Limit traffic to 10G on incoming interface. Drop excess traffic

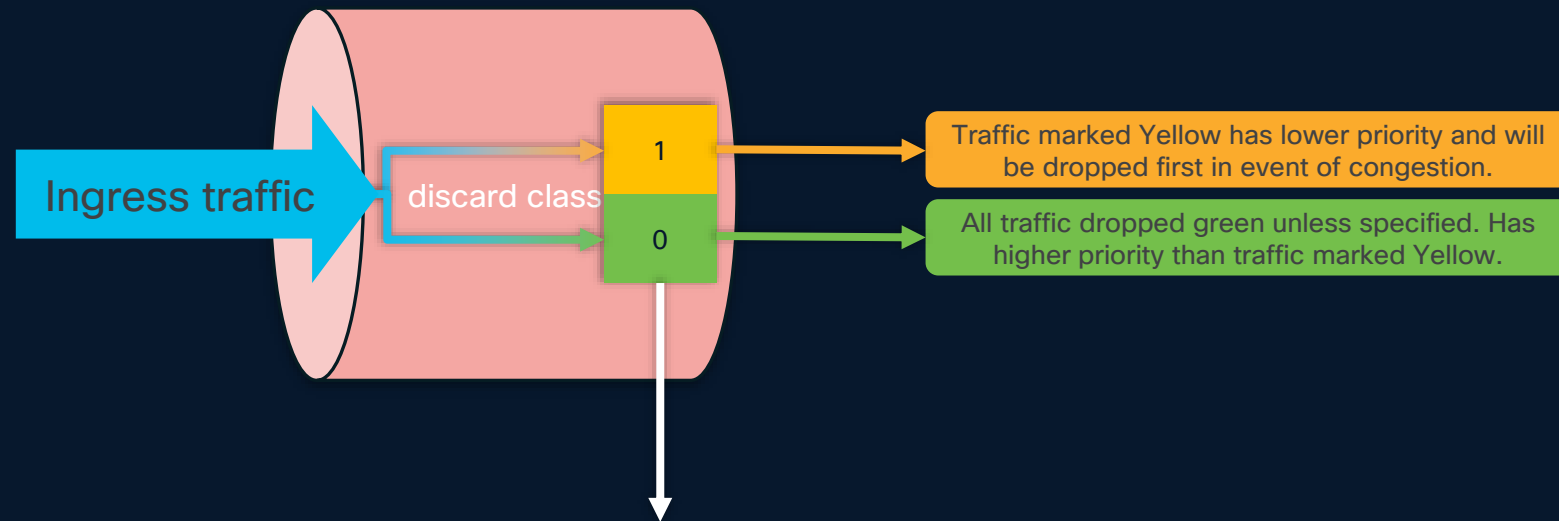
Egress shaping

```
policy-map type queueing test-shape
class tc3
  shape average percent 10
```

Limit traffic to 10% of link speed on outgoing interface. Buffer excess traffic on the **ingress interface.**

Congestion management

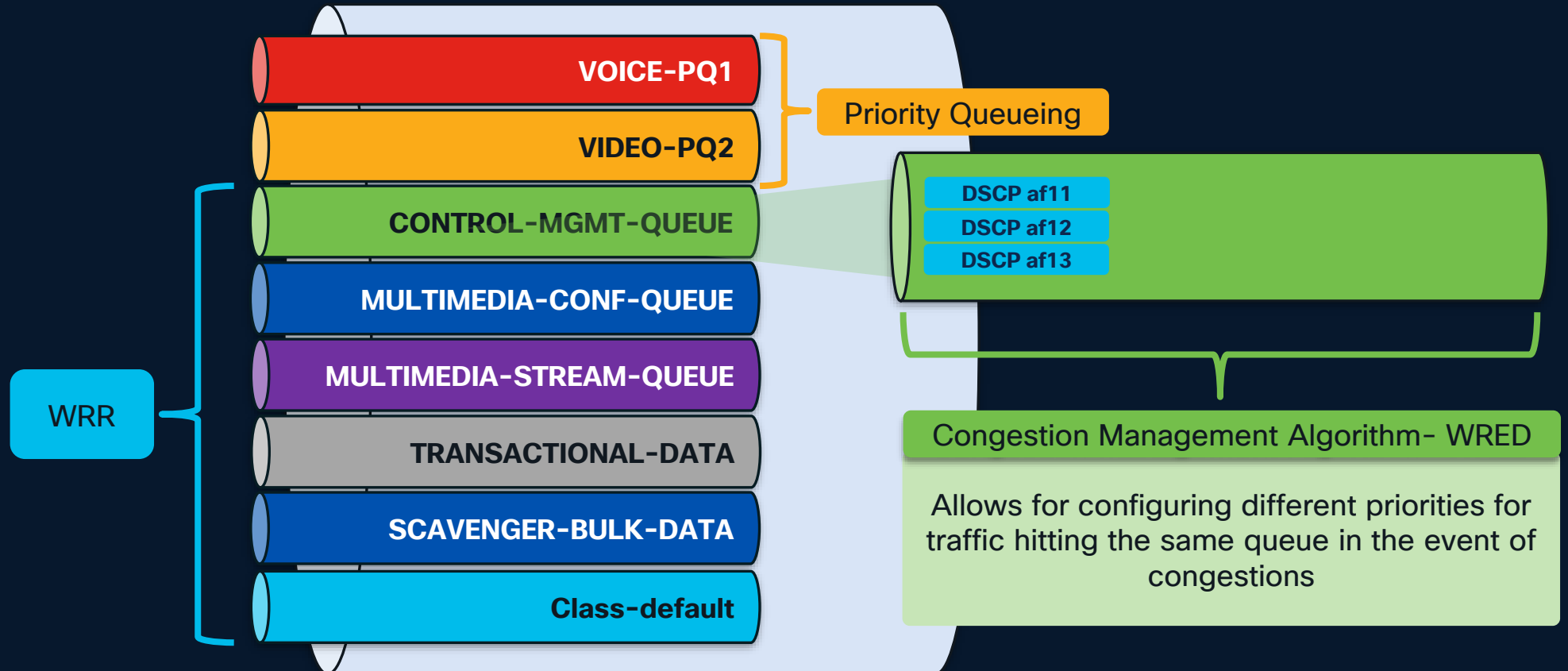
Using traffic-color



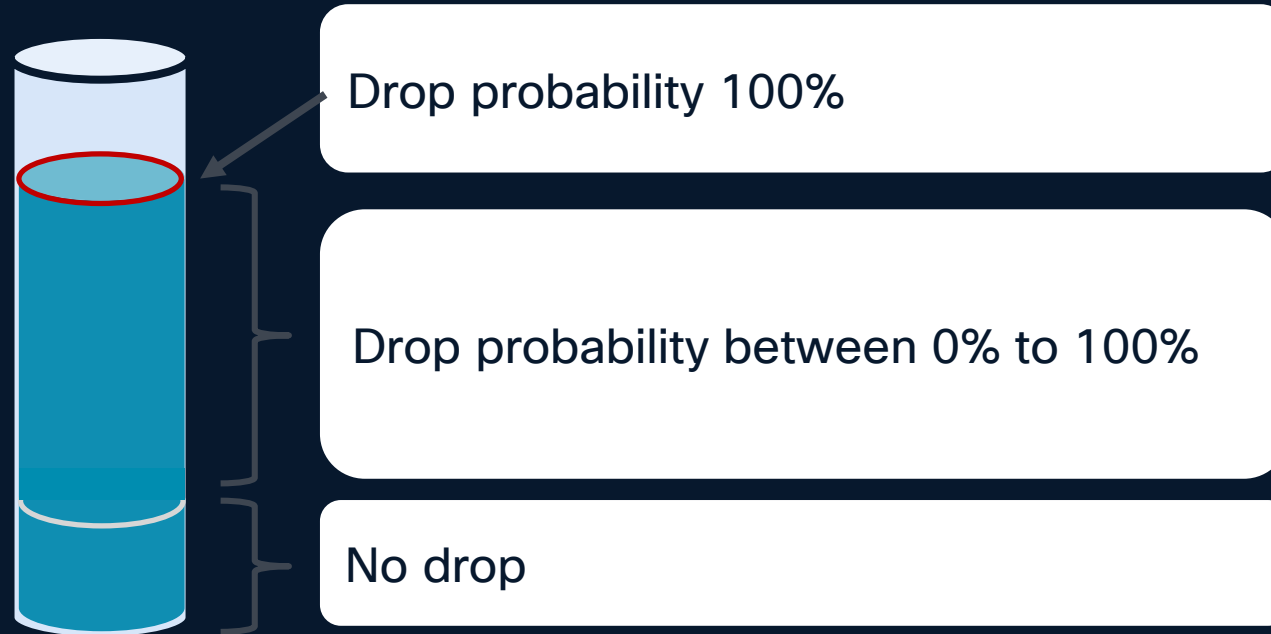
- Discard class is a 1-bit field
- Two possible values – 0 (**Green**) and 1 (**Yellow**)

Congestion management

Where do we need it?



Weighted Random Early Drop (WRED)



Congestion management

Config example

```
class-map match-all REALTIME
  match dscp ef
class-map match-all default-green
  match dscp af11
```

```
policy-map INGRESS
  class REALTIME
    set traffic-class 7
  class default-green
    set traffic-class 0
  class class-default
    set discard-class 1
```

```
Policy-map type queueing EGRESS
class-map tc7
  match traffic-class 7

policy-map type queueing EGRESS
  class tc7
    priority level 1
  class class-default
    random-detect discard-class-based
    random-detect discard-class 0 percent 80 100
    random-detect discard-class 1 percent 40 80
```

Default-green class (matching af11) is sent to tc0. By default, it is colored green.

Other traffic which hits tc0 is manually colored yellow.

When there is congestion, traffic colored yellow is dropped first.

We start dropping yellow packets at 40% of queue utilization.
At **80% queue utilization**, all yellow packets are dropped, and we start dropping **green**.

Egress Marking

```
Class-map match-any Servers
  match access-group name Server_IPv4
```

```
Class-map match-any IOT-Traffic
  match vlan 10
```

```
Class-map match-any Webex-traffic
  match protocol webex
```

```
Policy-map ingress-marking
```

```
Class Servers
  set traffic-class 6
  set qos-group 6
```

```
Class IOT-Traffic
  set traffic-class 5
  set qos-group 5
```

```
Class Webex-traffic
  set traffic-class 4
  set qos-group 4
```

```
Interface gi1/0/1
  service-policy input ingress-marking
```

```
class-map match-any dscp-af41
  match qos-group 6
```

```
!
Class-map match-any dscp-af11
  match qos-group 5
```

```
policy-map egress-map
```

```
class dscp-af41
  set dscp af41
```

```
class dscp-af11
  set dscp af11
```

```
!
```

```
interface interface gi1/0/3
```

```
  service-policy type queueing output egress-queueing
  service-policy output egress-map
```

Change packet tags as if egresses the switch. Use this to **re-mark untagged traffic** before it is sent upstream.

Buffers

UADP QoS Recap

Buffers

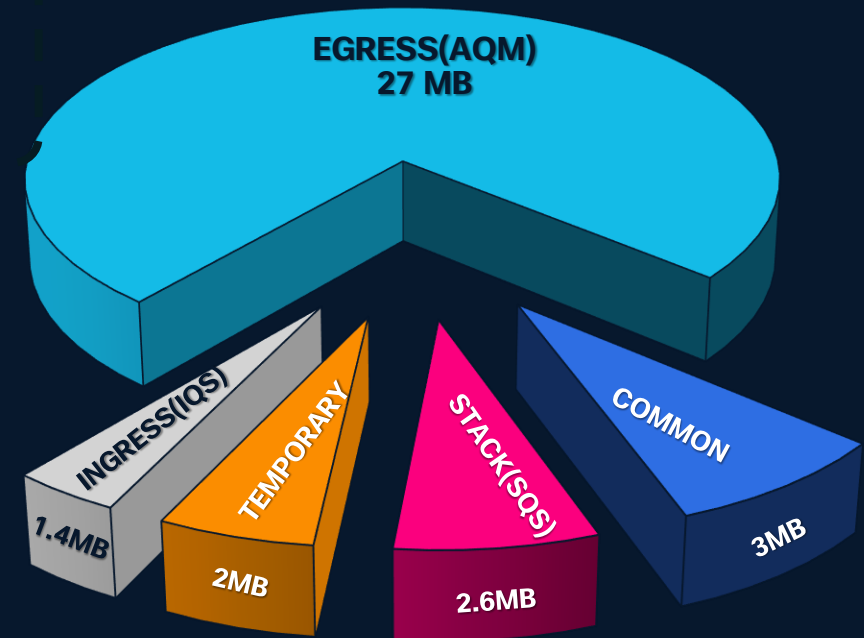
- Dedicated and shared: use dedicated first then shared
- Dynamic Threshold Scale (DTS): Algorithm to managed the shared buffer

Dedicated

- Allocated to each port on boot.
- Cannot be dynamically changed/edited

Shared

- Dynamically assigned to ports for burst absorption.
- Returned to common pool when not in use.

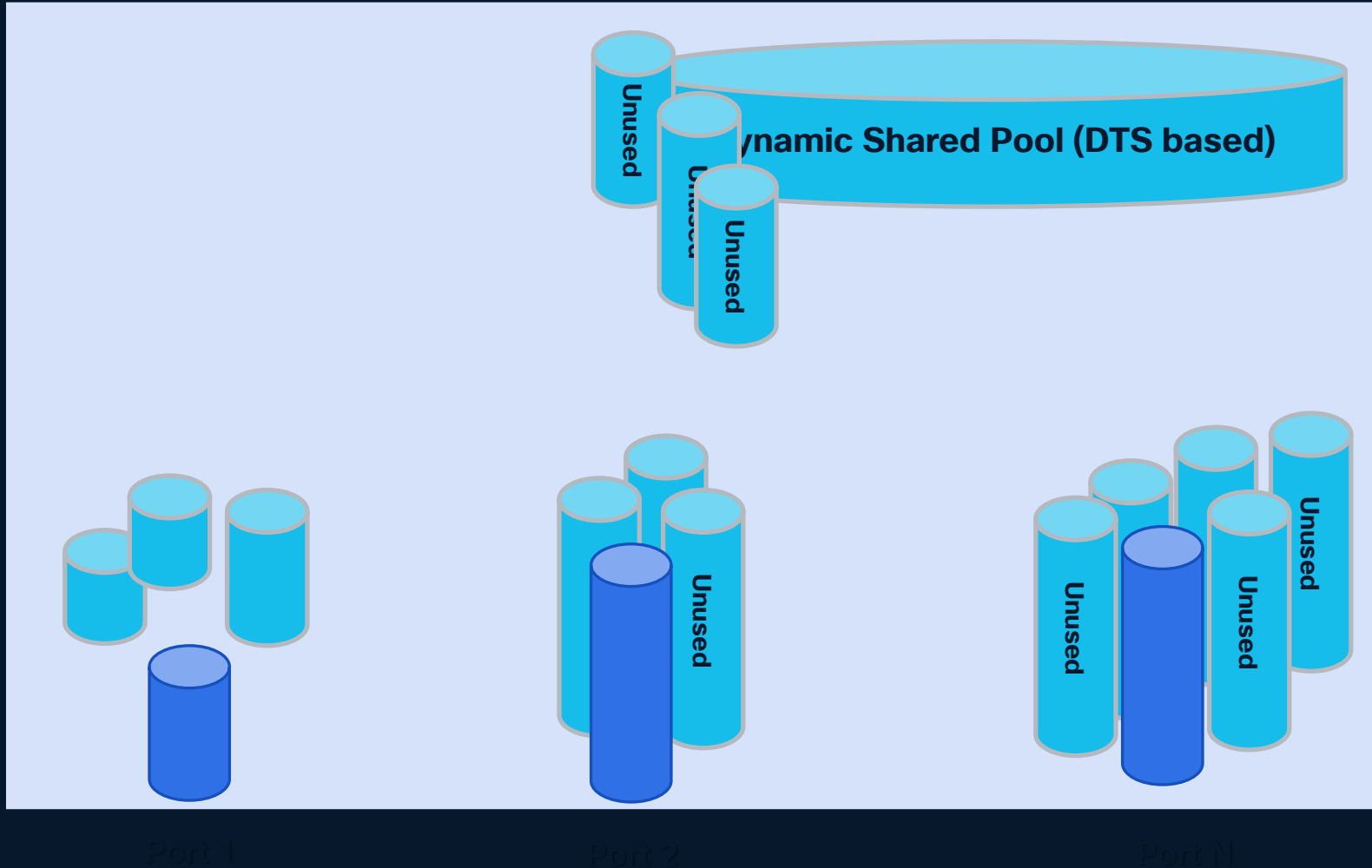


UADP 3.0

- UADP 3.0 specific
 - Buffer can be shared across two cores
 - “qos share-buffer” to enable the unified buffer

UADP QoS Recap

Shared Buffers

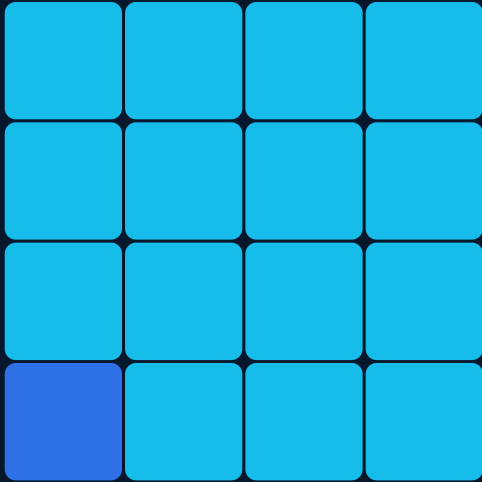


- Shared buffer is good for burst absorption.
- Dedicated buffer is good for predicated performance for each port.
- Buffer management is flexible: Dedicated plus shared.
- Configurable dedicated threshold per port/queue
- Configurable global maximum shared threshold
- Automatically adjusted depends on the available shared pool

UADP QoS recap – Configuring the softmax multiplier

What value to use?

qos queue-softmax-multiplier 100



I can give you maximum 100% of what you have

Shared Pool

Dedicated



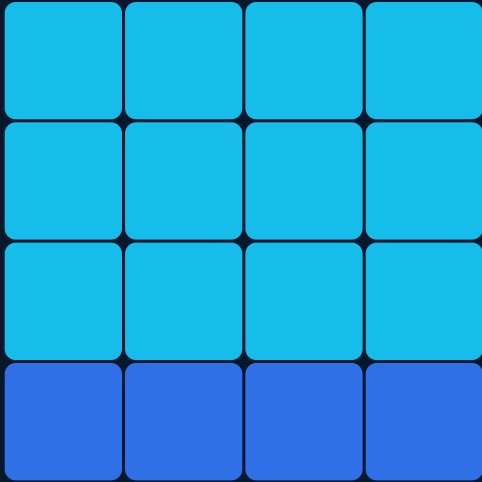
Interface 1/0/1

I need additional buffers to handle this traffic

UADP QoS recap – Configuring the softmax multiplier

What value to use?

qos queue-softmax-multiplier 400



I can give you maximum 400% of what you have

Shared Pool

Dedicated



Interface 1/0/1

I need additional buffers to handle this traffic

UADP QoS

Tweaking the Buffers

qos queue-softmax-multiplier X

Increases/decreases max buffers equally between all queues (apart from pq1)

Queue buffers ratio x

Granular control of for specific queues (used within a class in a policy)

By default, buffers are distributed proportionally between queues based on weight.

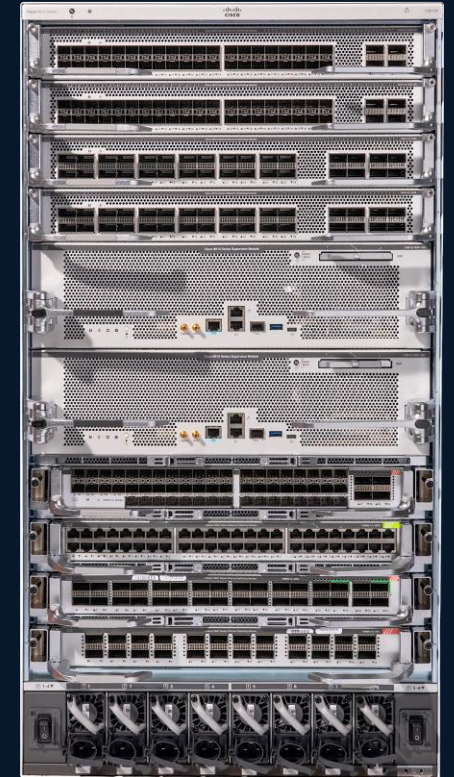
Silicon One Buffers

Cisco C9350 Switches



C9350 Series – 18MB buffer

Cisco C9610 Switches



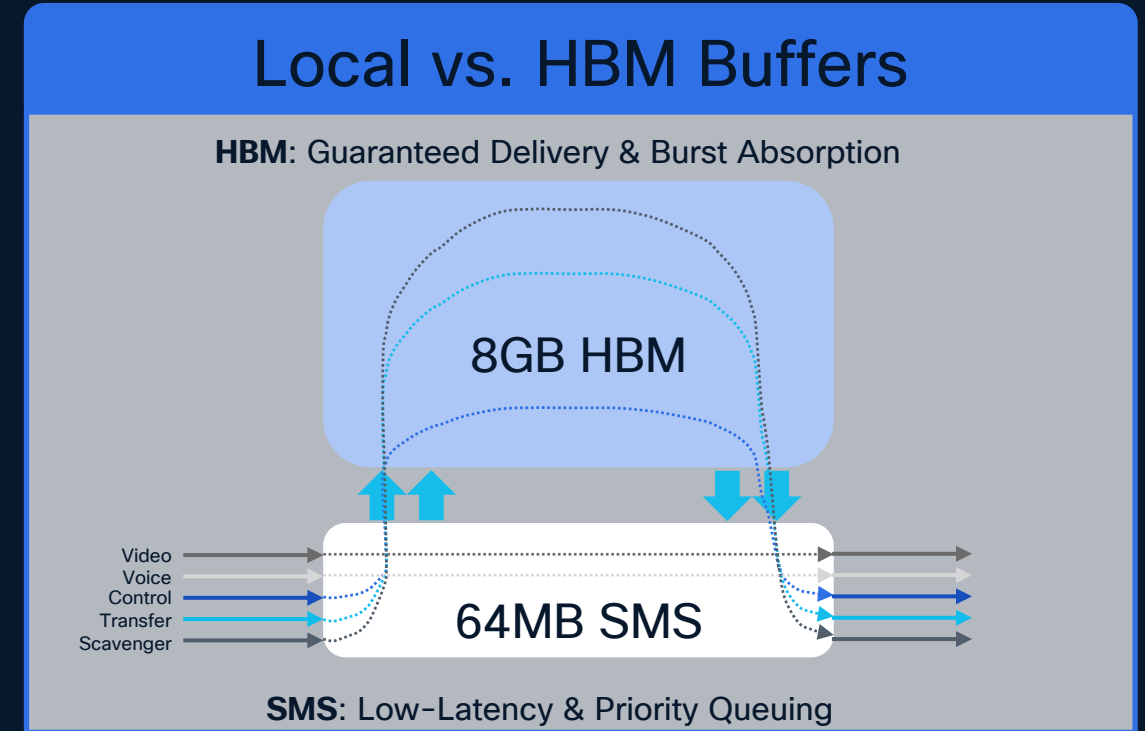
C9610-SUP3 – 64MB buffer

C9610-SUP3XL – 64MB buffer with expandable 8GB

Silicon One – K100 buffers

C9610-SUP3XL

- Two different buffers to address two different requirements.
 - 64 MB of **Shared Memory Sub-system** (SMS) buffers:
 - Low latency packet queueing (video/voice packets)
 - Shallow specialized pool of buffers for quick queueing.
 - 8 GB of **High Bandwidth Memory** (HBM) buffers:
 - Deep pool of on-demand buffers for guaranteed delivery.
 - Reserve to absorb occasional bursts or address speed over-subscription between ingress and egress.



- Packet will always hit the SMS buffers first.
- SMS send the packet to HBM if additional buffers are needed.
- HBM **CANNOT** send the packet to the output queue, it has to be sent to the SMS again to be sent to the egress.

The background of the slide is a dark blue field filled with dynamic, flowing light streaks. These streaks are primarily in shades of bright blue and cyan, with some warmer orange and pinkish tones visible in the upper right quadrant. The lines are curved and layered, creating a sense of depth and movement, reminiscent of light trails or data flow.

Putting everything together

Config Considerations

1 Define the problem/behavior addressed with QoS.

Simply copy-pasting existing configs between platform families will always throw errors due to differences in syntax and supported actions between platforms.

2 How many times do you want to split your traffic – Upto 8 queues possible with our ASIC

Its often not as much as you think you need. Broad generalized splits often are more efficient than granular splits

3 Do you want multiple strict priority classes?

Know what strict means. All traffic coming into it will be serviced at the expense of other classes.

4 Define traffic shaping/policing or sharing between queues.

Police/shape priority queues. Use weights to control bandwidth sharing with remaining queues

5 Do you want differential priority for packets within same class? – Use WRED

Advanced configuration options, not required for most use cases.

Putting it all together

Building a Silicon One Q200 configuration - 2P2Q2T Example

- I want 2 priority queues
 - Map tc7 and tc6 to priority levels 1 and 2.
- 2 normal queues
 - Create a custom queue (use anything between tc5 to tc1) and class-default (tc0)
- Custom queue must be higher priority than other. (roughly 2:1 priority)
 - Use 'bandwidth percent' command to assign weights.
- I have mix of traffic coming into my custom queue and I want to differentiate between them.
 - Use WRED with traffic-color. I must color on the ingress.

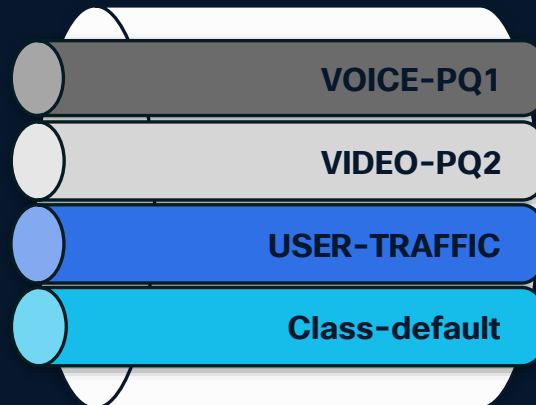
Putting it all together

Building a Silicon One Q200 configuration - 2P2Q2T Example

```
class-map match-any VOICE-PQ1
  match dscp ef
class-map match-any VIDEO-PQ2
  match dscp 24
class-map match-any USER-TRAFFIC
  match dscp cs3 af31 af32
Class-map match-any USER-TRAFFIC-YELLOW
  match dscp af33

policy-map CAMPUS_INGRESS_POLICY
  class VOICE-PQ1
    set traffic-class 7
  class VIDEO-PQ2
    set traffic class 6
  class USER-TRAFFIC
    set traffic class 5
  Class USER-TRAFFIC-YELLOW
    set traffic class 5
    set discard-class 1

Interface gig1/0/1
  service-policy input CAMPUS_INGRESS_POLICY
```



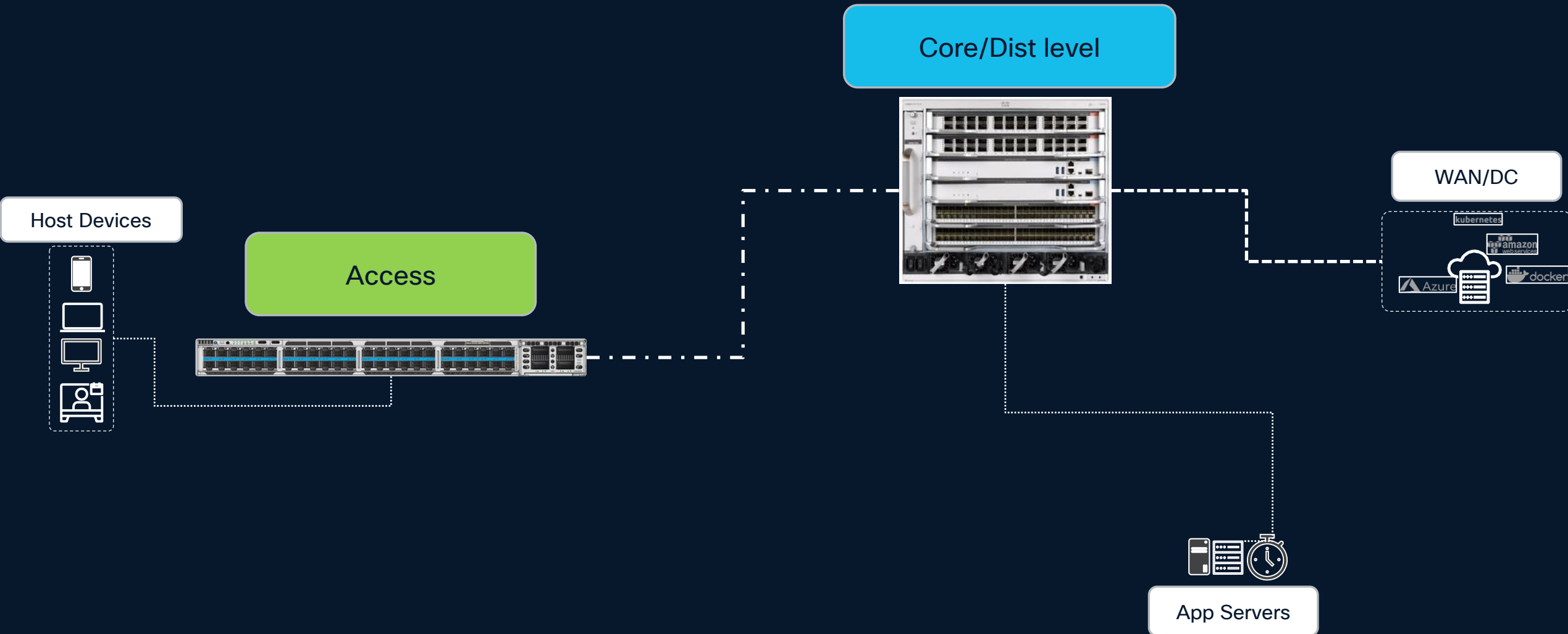
```
class-map match-any tc7
  match traffic-class 7
class-map match-any tc6
  match traffic-class 6
class-map match-any tc5
  match traffic-class 5

policy-map type queueing CAMPUS_EGRESS_POLICY
  class tc7
    priority level 1
  class tc6
    priority level 2
  class tc5
    bandwidth percent 60
    random-detect discard-class-based
    random-detect discard-class 0 percent 80 100
    random-detect discard-class 1 percent 40 100
  class class-default
    bandwidth percent 30

Interface gig1/0/1
  service-policy type queueing output CAMPUS_EGRESS_POLICY
```

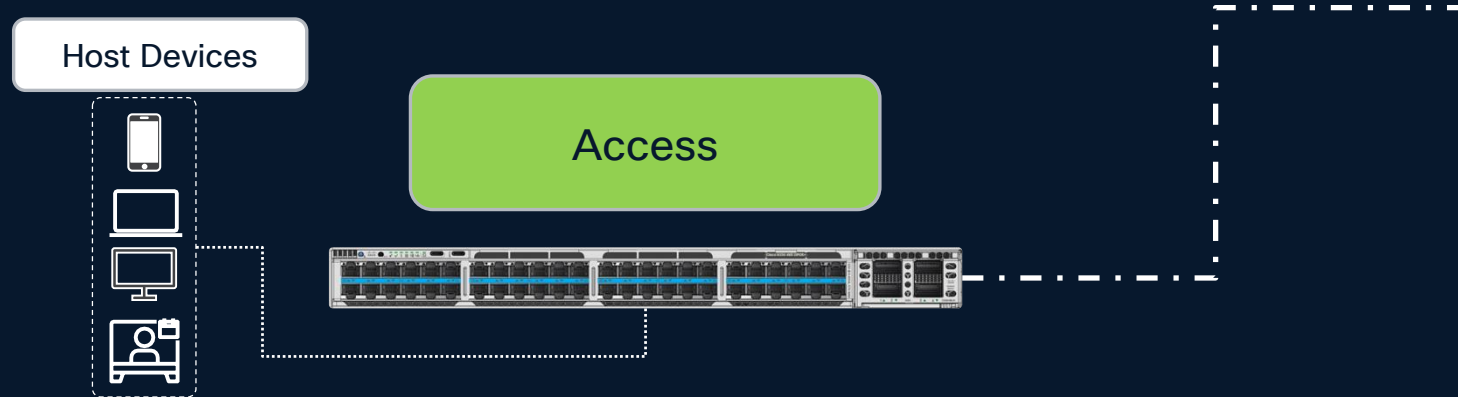
QoS design and config considerations

How to approach QoS design at a network level



QoS design and config considerations

How to approach QoS design for an access switch



Goal at access is to identify and mark end host traffic. Core/Dist will leverage markings made here.

Low-priority traffic (like YouTube) **can be limited close to the host.**

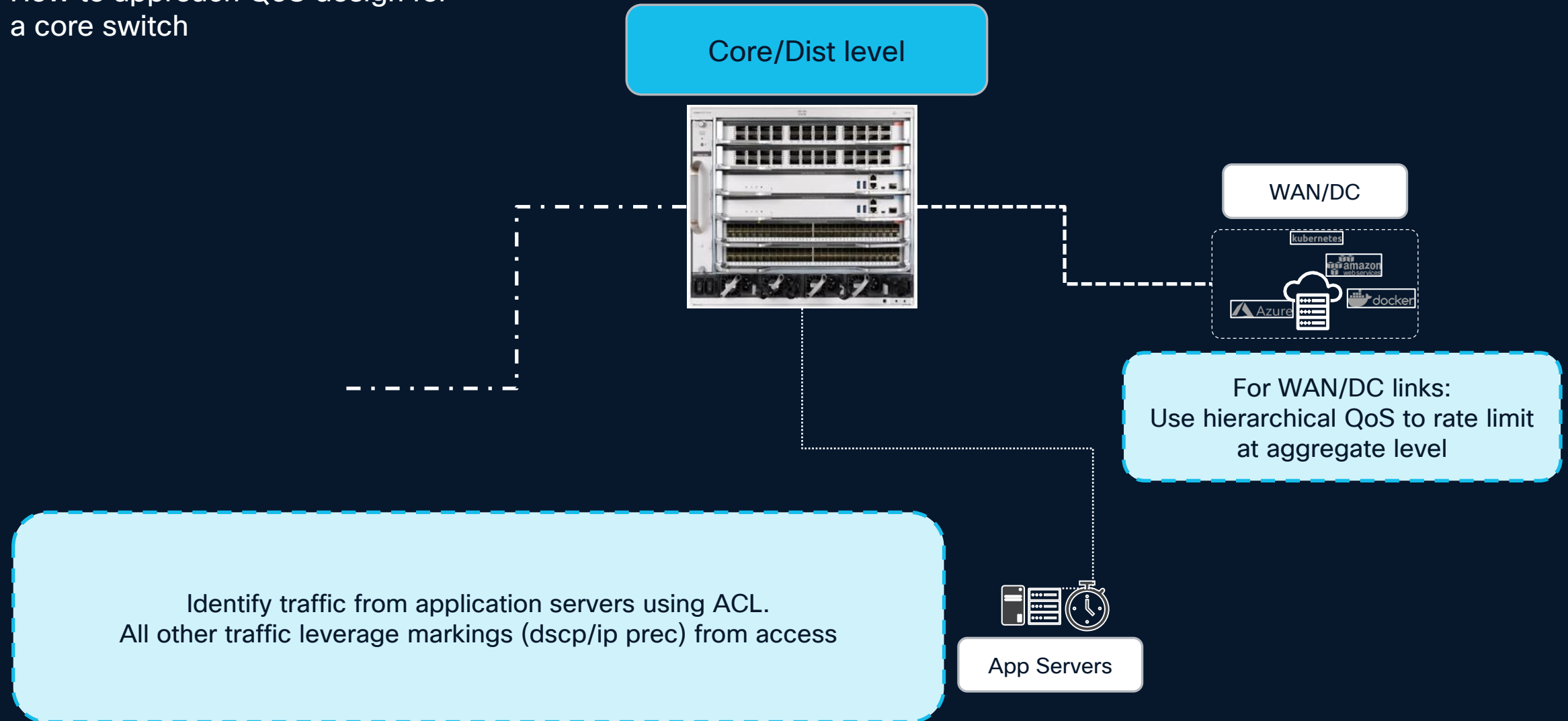
Use ACLs/VLAN/Protocols to match untagged traffic **as it comes to the switch**

Ensure packets are tagged so core/dist switches match the tags.

Split into up-to 8 queues on the egress.

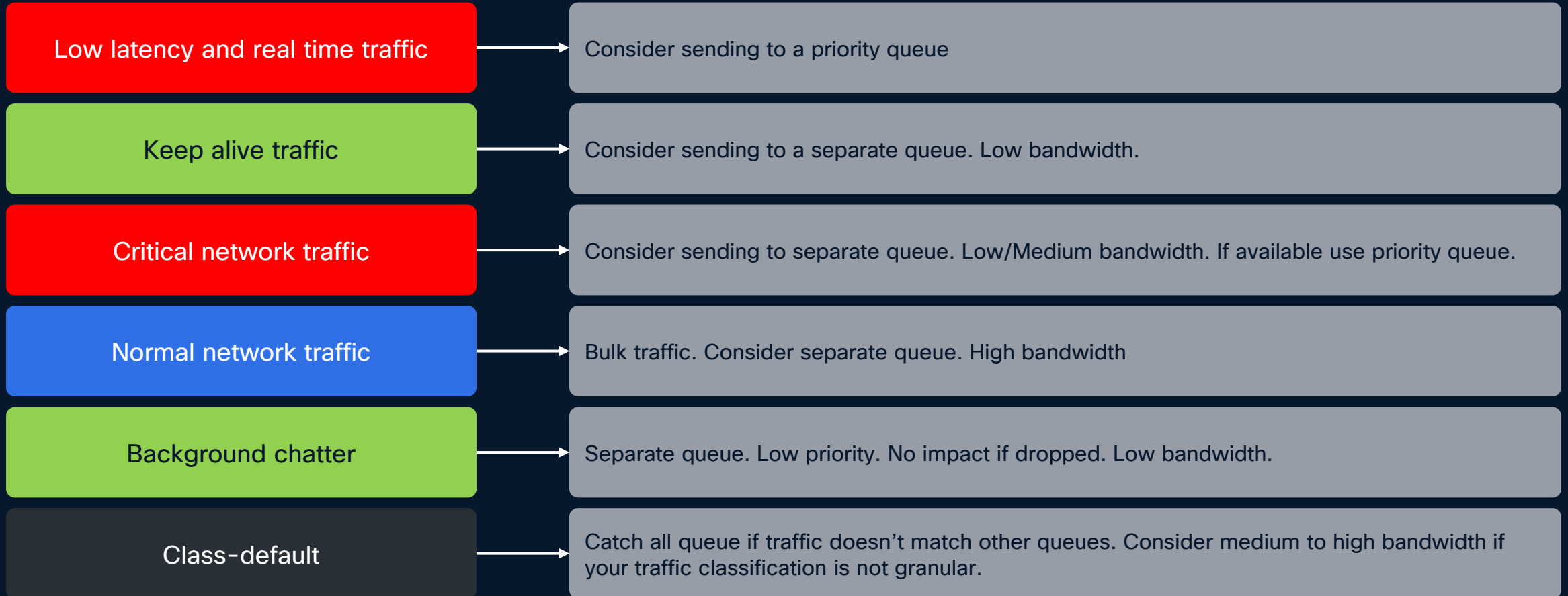
QoS design and config considerations

How to approach QoS design for a core switch



How to design a QoS config

How to classify traffic



Keep it consistent. 8 queues at access = 8 queues are core

Config migrations

Config migration

Examples

Catalyst 6K Configuration

```
class-map type lan-queuing match-all REALTIME
  match dscp ef
class-map type lan-queuing match-all NETWORK_CONTROL
  match dscp cs6 cs7
class-map type lan-queuing match-all VIDEO
  match dscp cs3 af31 af32 af33
...
policy-map type lan-queuing CAMPUS_EGRESS_6800_POLICY
  class type lan-queuing REALTIME
    priority level 1
  class type lan-queuing NETWORK_CONTROL
    bandwidth remaining percent 10
  class type lan-queuing VIDEO
    bandwidth remaining percent 20
  class class-default
    random-detect dscp-based
    random-detect dscp af11 percent 80 100
```

1. Classified Based on DSCP value
2. 4 classes (3 defined + default)

3. 4 queues
4. 1 priority queue
5. Scheduling is WRR with “bandwidth remaining”
6. Congestion management is WRED with the default class

Config migration

Examples

Catalyst 6K Configuration

```
class-map type lan-queuing match-all REALTIME
  match dscp ef
class-map type lan-queuing match-all NETWORK_CONTROL
  match dscp cs6 cs7
class-map type lan-queuing match-all VIDEO
  match dscp cs3 af31 af32 af33
...
policy-map type lan-queuing CAMPUS_EGRESS_6800_POLICY
  class type lan-queuing REALTIME
    priority level 1
  class type lan-queuing NETWORK_CONTROL
    bandwidth remaining percent 10
  class type lan-queuing VIDEO
    bandwidth remaining percent 20
  class class-default
    random-detect dscp-based
    random-detect dscp af11 percent 80 100
```

1. Classified Based on DSCP value
2. 4 classes (3 defined + default)
3. 4 queues (traffic-class), traffic-7 is priority level 1



Apply policy on the ingress interface

Catalyst 9K Configuration

```
class-map match-all REALTIME
  match dscp ef
class-map match-all NETWORK_CONTROL
  match dscp cs6 cs7
class-map match-all VIDEO
  match dscp cs3 af31 af32 af33
class-map match-all default-green
  match dscp af11

policy-map INGRESS
  class REALTIME
    set traffic-class 7
  class NETWORK_CONTROL
    set traffic-class 6
  class VIDEO
    set traffic-class 5
  class default-green
    set traffic-class 0
  class class-default
    set discard-class 1
```

Note: class-default is always assigned with traffic-class 0

Config migration

Examples

Map the traffic-class marking defined on the ingress

Catalyst 6K Configuration

```
policy-map type lan-queuing CAMPUS_EGRESS_6800_POLICY
  class type lan-queuing REALTIME
    priority level 1
  class type lan-queuing NETWORK_CONTROL
    bandwidth remaining percent 10
  class type lan-queuing VIDEO
    bandwidth remaining percent 20
  class class-default
    random-detect dscp-based
    random-detect dscp af11 percent 80 100
```

3. 1 priority queue
4. Scheduling is WRR with “bandwidth remaining”
5. Congestion management is WRED with the default class

Apply policy on the egress interface

Catalyst 9K(Q200) Configuration

```
class-map tc7
  match traffic-class 7
class-map tc6
  match traffic-class 6
class-map tc5
  match traffic-class 5

policy-map type queuing EGRESS
  class tc7
    priority level 1
  class tc6
    bandwidth percent 1
  class tc5
    bandwidth percent 2
  class class-default
    random-detect discard-class-based
    random-detect discard-class 0 percent 80 100
    random-detect discard-class 1 percent 40 100
```

Note: C6K WRED default min is 40, max is 100.

Closing

Comparison

Features	UADP ASIC	Silicon One (Q200)	Silicon One (A100)	Silicon One (K100/E100)
Trust	Trust all ports by default	Trust all ports by default	Trust all ports by default.	Trust all ports by default.
Conditional trust	Supported	Not supported	Supported	Supported
Classification	Based on Packet header and ACL for ingress and packet-header/qos-groups for egress	Based on packet header and ACL for ingress Based on traffic-class for egress	Based on packet header and ACL for ingress Based on traffic-class for egress	Based on packet header and ACL for ingress Based on traffic-class for egress
Marking	Header, Table-map, QoS-Group for ingress Header and table-map for egress	Header, Table-map, QoS-group, traffic-class , discard-class for ingress Header and table-map for egress	Header, Table-map, QoS-group, traffic-class , discard-class for ingress Header and table-map for egress	Header, Table-map, QoS-group, traffic-class , discard-class for ingress Header and table-map for egress
Marking number of queues	32	32	32	32
Policing	Both ingress and egress	Ingress only	Ingress only	Ingress only

Comparison

Features	UADP ASIC	Silicon One ASIC (Q200)	Silicon One A100	Silicon One (K100/E100)
Queueing	Based on header or QoS group Bandwidth and Bandwidth remaining	Based on traffic-class Bandwidth remaining	Based on traffic-class Bandwidth percent	Based on traffic-class Bandwidth percent
Queueing number of queues	8	8	8	8
Buffering	Dedicated and shared buffer with DTS	SMS: Low-latency & priority queueing HBM: Guaranteed Delivery & Burst Absorption	SMS – 18 MB	SMS – 64 MB HBM – 8 GB (on K100 only)
Shaping	Egress	Egress	Egress	Egress
Congestion Management	WTD: three thresholds per class WRED: three thresholds' pairs per class	WTD: Not supported WRED: two thresholds' pairs per class	WTD: Not supported WRED: Two threshold pairs per class (FCS+ considering 3 for select platforms)	WTD: Not supported WRED: Two threshold pairs per class

Comparison

Features	UADP ASIC	Silicon One ASIC (Q200)	Silicon One A100	Silicon One K100/E100
MAC QoS ACL at ingress	Supported	Unsupported	Supported	Supported
Match L2 fields using ACL for L3 packets	Supported	Unsupported	Unsupported	Unsupported
SVI Policy	Supported (Marking only)	Unsupported	Supported (Marking only)	Supported (Marking only)
NetFlow QoS (AVC)	Supported	Unsupported	Supported – FCS+	Unsupported
Policy on port-channel	Unsupported	Supported	Supported	Supported
Sub-interface queueing	Unsupported – Main interface only	Supported on L3 interfaces	Supported on L3 interfaces	Supported on L3 interfaces
Sub-interface (number of queues)	NA	2	8	8

Complete Your Session Evaluations



Complete a minimum of 4 session surveys and the Overall Event Survey to be entered in a drawing to win 1 of 5 full conference passes to Cisco Live 2026.



Earn 100 points per survey completed and compete on the Cisco Live Challenge leaderboard.



Level up and earn exclusive prizes!



Complete your surveys in the Cisco Live mobile app.

Continue your education



Visit the Cisco Showcase for related demos



Book your one-on-one Meet the Engineer meeting



Attend the interactive education with DevNet, Capture the Flag, and Walk-in Labs



Visit the On-Demand Library for more sessions at www.CiscoLive.com/on-demand

Contact me at: ndiwakar@cisco.com

Thank you

CISCO Live !

