Considerations on stream aggregation in enhanced CQF scheme

Chen Shuang, Wang Tongtong
Huawei Technologies Co. Ltd

2022/9/16

IEEE 802.1 TSN Meeting

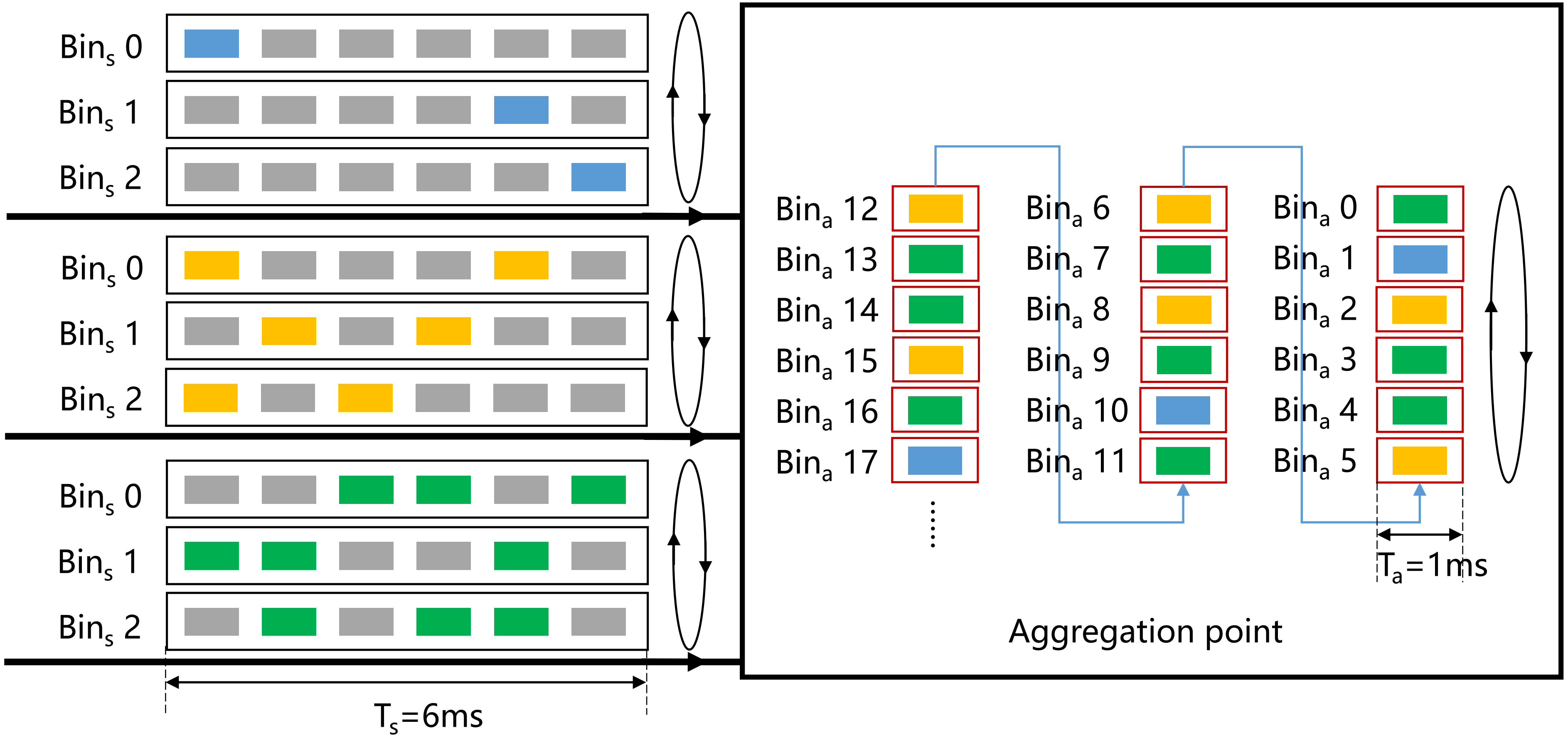
Purpose of this presentation

- These slides aims to present some considerations of stream aggregation for enhanced CQF, which is proposed by Norman Finn in July for the PAR of IEEE P802.1Qdv Enhancements to Cyclic Queuing and Forwarding[1]
- These slides include several instances to show how to operate during aggregation/dis-aggregation, by considering enhanced CQF of both time-based(CQF-N) as well as count-based(Paternoster), the scheduling details are given, the buffer and one-hop delay analyses are also proposed
- The brief idea of various cycles for stream aggregation is also proposed

[1] Norman Finn, dv-finn-CQF-stream-aggregation-v01, July, 2022

Recap: stream aggregation for enhanced CQF

- "Stream Aggregation is the treatment, for at least the purpose of QoS, of all frames belonging to 1 or more TSN (Time-Sensitive Networking) Streams as if they belong to one single Stream" [1]



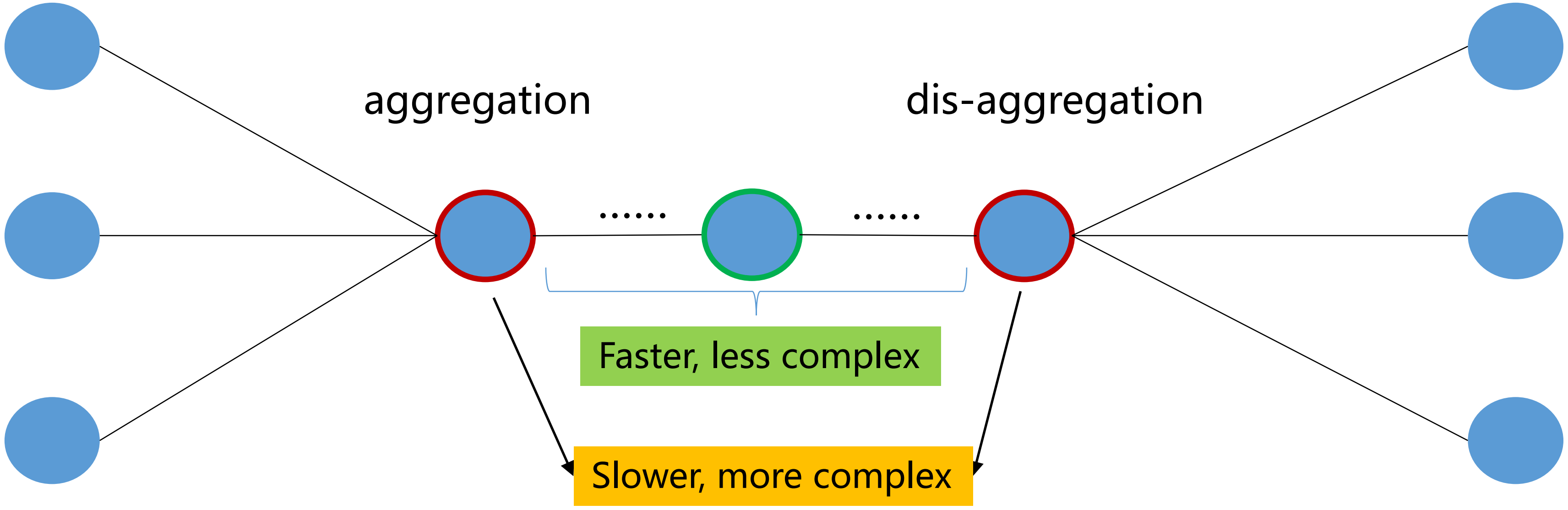
- 3 streams are regarded as one single stream on the output port
- Each bin is 6X faster on output port, therefore per-hop delay can be smaller
- Packets should be carefully dispersed in the new bins, to **avoid collisions** among 3 arriving streams.
- Meanwhile, **original characteristics should be recovered** for component streams at disaggregation point

Straightforward serialization is an option at aggregation point, but it will make stream characteristics recovery difficult at dis-aggregation point. Meanwhile, the resulting buffer and delay are not clear in this way

[1] Norman Finn, dv-finn-CQF-stream-aggregation-v01, July, 2022

Repoint: benefits and costs of aggregation for enhanced CQF

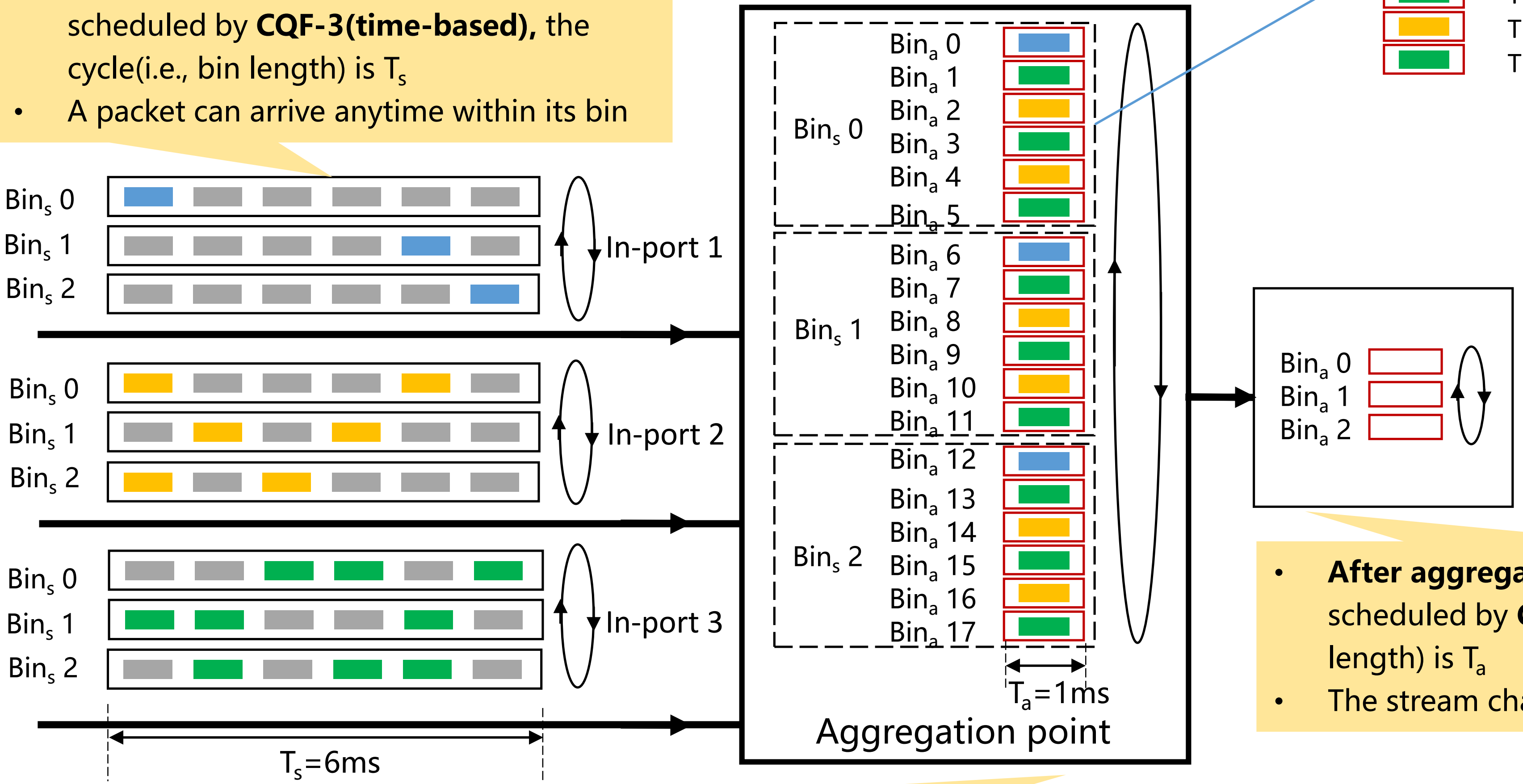
- **benefits**
  - After Aggregation, the stream identification as well as per-stream state machines are reduced
  - After aggregation, the per-hop buffer and per-hop delay for component streams are reduced
- **costs**
  - Extra buffer and delay are introduced at aggregation/disaggregation point
  - Extra designing complexity are needed at aggregation/dis-aggregation point



Whether to apply stream aggregation or not, deeply depends on the network topology, paths of streams, network resources and stream latency requirements, the overall tradeoff should be carefully considered

Time-based case: aggregation point

- **Before aggregation**, each single stream is scheduled by **CQF-3(time-based),** the cycle(i.e., bin length) is T_s
- A packet can arrive anytime within its bin



🟦 The 1st packet in a cycle from in-port 1: offset(1,1) = 0
🟩 The 1st packet in a cycle from in-port 3: offset(3,1) = 1
🟨 The 1st packet in a cycle from in-port 2: offset(2,1) = 2
🟩 The 2nd packet in a cycle from in-port 3: offset(3,2) = 3
🟨 The 2nd packet in a cycle from in-port 2: offset(2,2) = 4
🟩 The 3rd packet in a cycle from in-port 2: offset(3,3) = 5

The regular scheduling rule

- **The regular scheduling rule is** to make a fixed packet placement with strict periodic pattern
- The scheduling rule can be computed locally at the aggregation point

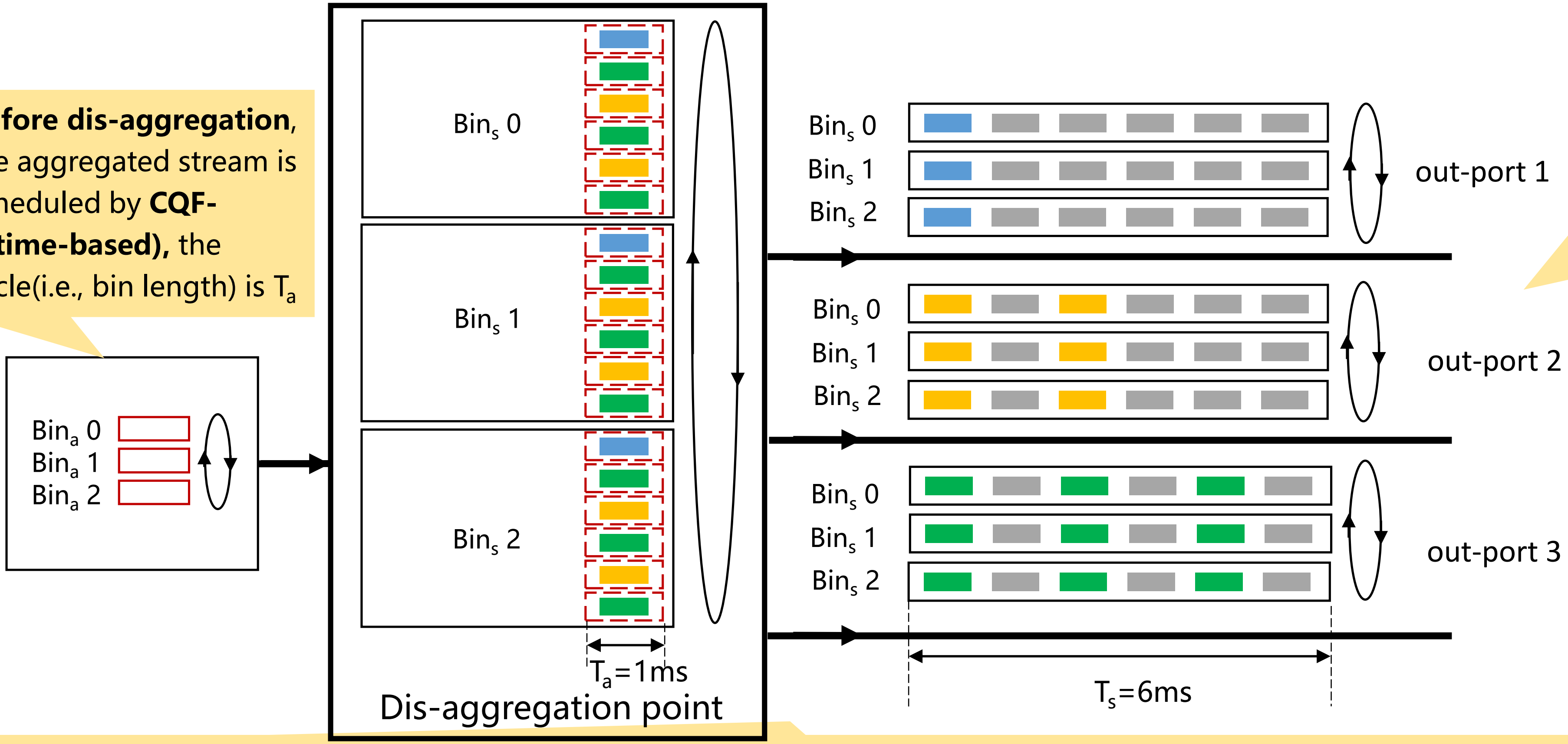
- **After aggregation**, the aggregated stream is also scheduled by **CQF-3(time-based)**, but the cycle(i.e., bin length) is T_a
- The stream characteristics can be **perfectly maintained**

- **At the aggregation point**, build up **3*6 T_a-length entity bins** in the output port, meanwhile consider **3 T_s-length virtual bins**, each of which maps 6 T_a-length entity bins
- For any packet, first determines the T_s-length virtual bin in the output port based on CQF-3, as if there is no T_a-length entity bins exist
- Then determines the certain T_a-length entity bin within the chosen T_s-length virtual bin, based on the byte counter of the packet in that virtual bin and a special **regular scheduling rule.**

- At aggregation point, **buffer is 3*M T_a-length entity bins (M=T_s/T_a, here M=6), per-hop delay is at most 3*T_s**

Time-based case: dis-aggregation point

- **Before dis-aggregation**, the aggregated stream is scheduled by **CQF-3(time-based),** the cycle(i.e., bin length) is T_a



- **After dis-aggregation**, the component stream has satisfied the original stream characteristics, which can be scheduled by **CQF-3(time-based)** again, the cycle(i.e., bin length) is T_s

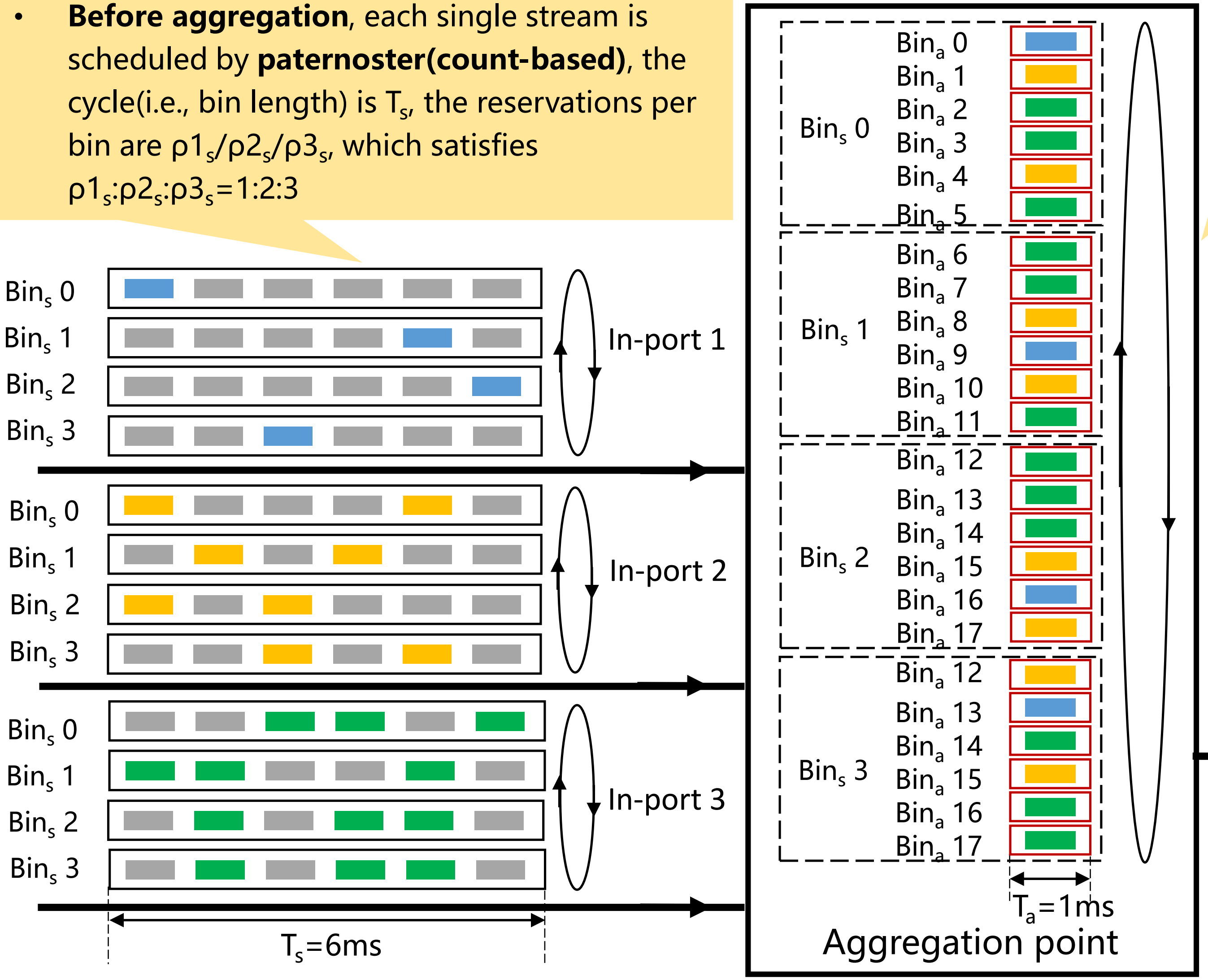
- **At the dis-aggregation point**, build up **3 T_s-length entity bins in each output port**
- The basic idea is to recovery the component streams back to their original stream characteristics. As the component streams have been regularly scheduled at aggregation point, we can directly determines the T_s-length entity bin to enter in the output port using CQF-3 based on the arriving time of each packet. In this way, the component stream characteristics can be well recovered
- **The only information needed at the dis-aggregation point is the original bin length T_s** of the component stream

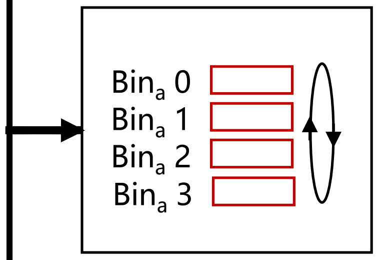
- At dis-aggregation point, **buffer is 3 T_s-length entity bins, per-hop delay is at most 3*T_s**

- **The rationality is to well maintain stream characteristics at aggregation point by using regular scheduling rules, to reduce the recovery complexity at dis-aggregation point, and there is no need to obtain original stream characteristic details from the source/aggregation point**

Count-based case: aggregation point

- **Before aggregation**, each single stream is scheduled by **paternoster(count-based)**, the cycle(i.e., bin length) is T_s, the reservations per bin are $\rho1_s/\rho2_s/\rho3_s$, which satisfies $\rho1_s:\rho2_s:\rho3_s=1:2:3$





- **At aggregation point**, build up **4*6 T_a-length entity bins** in the output port, set the reservation per entity bin as $\rho_a = \rho1_s = \rho2_s/2 = \rho3_s/3$, meanwhile **set the reservation per 6 entity bins as $\rho1/\rho2/\rho3$ for in-port 1/2/3**
- Meanwhile consider **4 T_s-length virtual bins**, each of which maps 6 T_a-length entity bins
- For any packet, determines the T_s-length virtual bin as well as T_a-length entity bin in the output port. Assume a packet from in-port Y arrives at X-th T_s-length cycle, the rule is:
  - If the unused reservation of virtual bin X+1 is enough, choose the nearest available entity bin(i.e., the unused reservation of this entity bin is enough) in virtual bin X+1, then entry into it
  - If not, judge whether unused reservation of virtual bin X+2 is enough. If yes, choose the nearest available entity bin in virtual bin X+2 and entry into it
  - If not, choose virtual bin X+3, and lastly choose virtual bin X+4
- 4*6 T_a-length entity bins is enough to ensure no packets are lost
- In this way, it can be naturally to distribute packets without collision

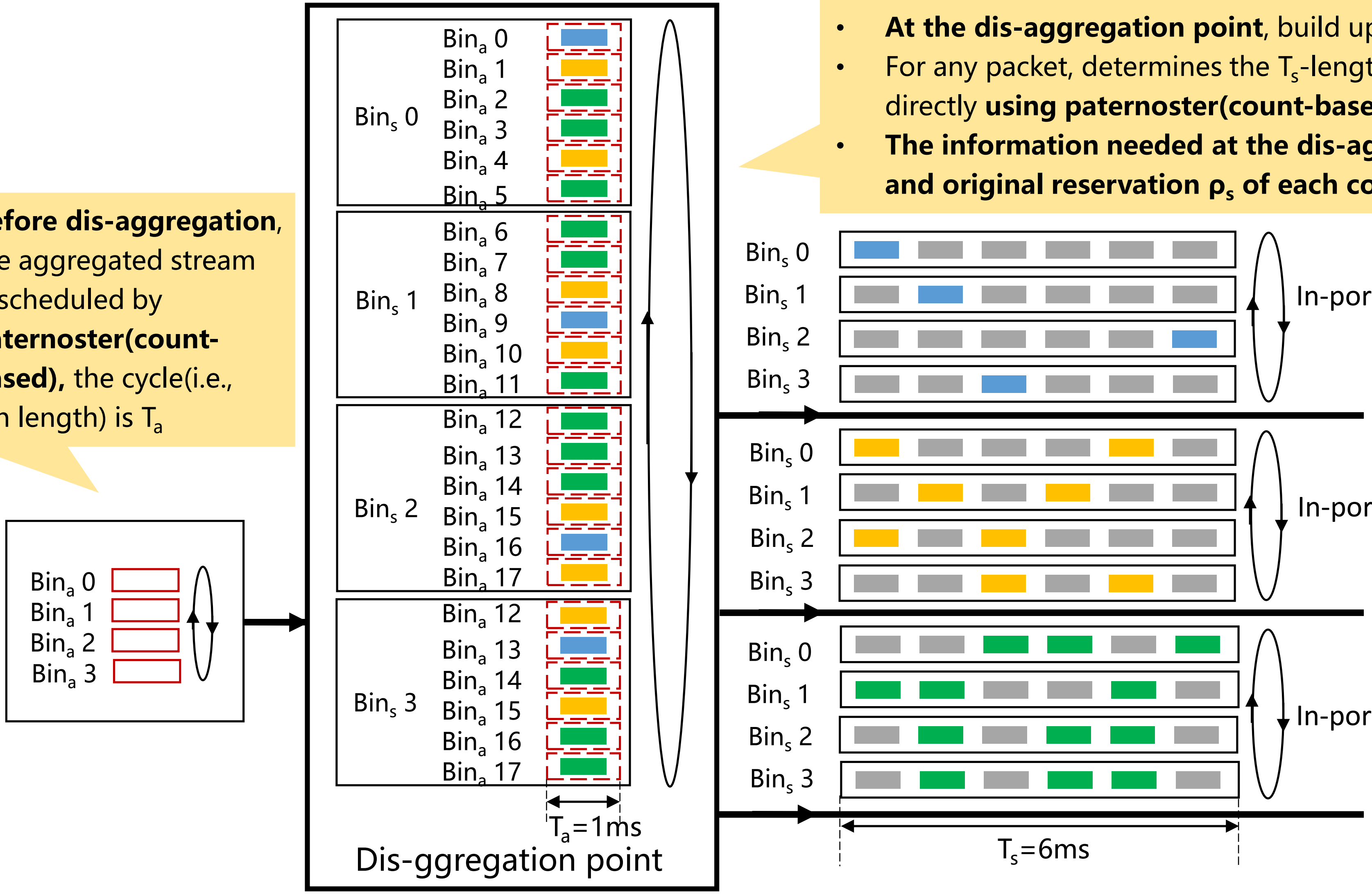
- **After aggregation**, the aggregated stream can be scheduled again with **paternoster(count-based)**, the cycle(i.e., bin length) is T_a , the reservation per bin is ρ_a

- At aggregation point, **buffer is 4*M T_a-length entity bins ($M=T_s/T_a$, here M=6), per-hop delay is at most 4*T_s**

Count-based case: dis-aggregation point

- **At the dis-aggregation point**, build up **4 T_s-length entity bins in each output port**
- For any packet, determines the T_s-length entity bin to enter in the output port by directly **using paternoster(count-based)**
- **The information needed at the dis-aggregation point is the original bin length T_s and original reservation ρ_s of each component stream**

- **Before dis-aggregation**, the aggregated stream is scheduled by **paternoster(count-based),** the cycle(i.e., bin length) is T_a



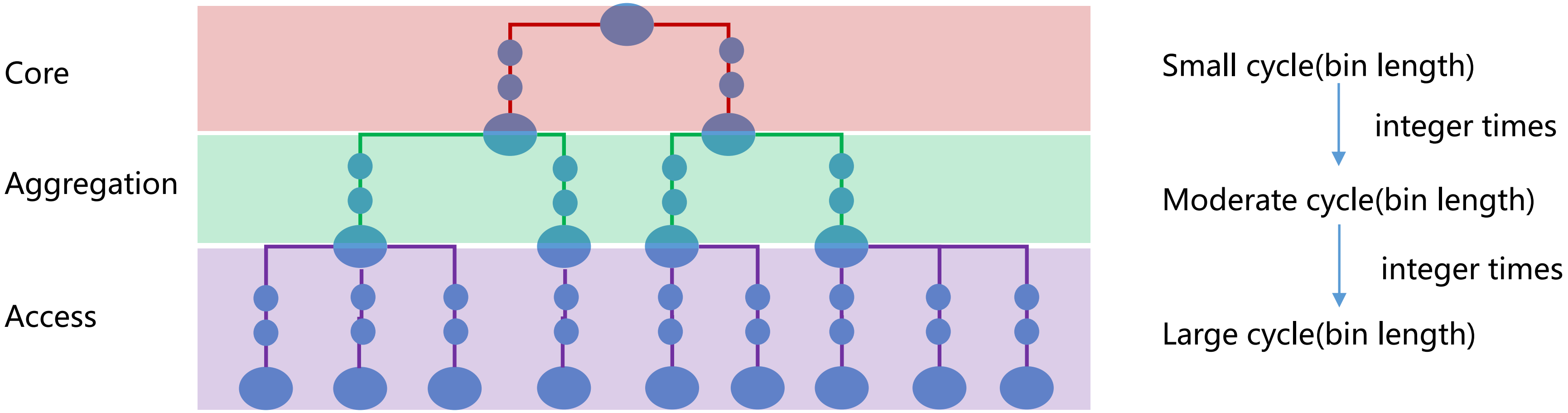
- **After dis-aggregation**, the component stream satisfies the original stream characteristics, which can be scheduled by **paternoster(count-based)** again, the cycle(i.e., bin length) is T_s

- **At dis-aggregation point, buffer is 4 T_s-length entity bins, per-hop delay is at most 4*T_s**

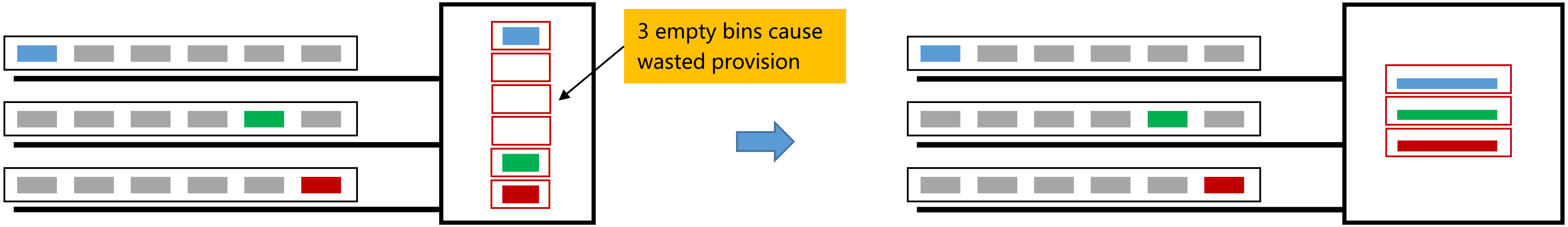
- **Again, stream characteristics are well maintained at aggregation point and after aggregation, so that the scheduling complexity is reduced and requirement of original stream characteristics is minimized at dis-aggregation point**

Think more: various cycles for aggregation

- Basic idea: consider different cycles(i.e., bin length) for aggregation in different areas of the network



- Save the bandwidth provision, bin numbers and scheduling complexity at aggregation/dis-aggregation points by choosing a more suitable bin length rather than the smallest one if possible



3 empty bins cause wasted provision

Packet size = 1500B
Bin length = 6ms
single stream Provision =
1500*8/6/1000=2Mbps

Packet size = 1500B
Bin length = 1ms
aggregation stream Provision =
1500*8/1/1000=12Mbps

Packet size = 1500B
Bin length = 6ms
single stream Provision =
1500*8/6/1000=2Mbps

Packet size = 1500B
Bin length = 1ms
aggregation stream Provision =
1500*8/2/1000=6Mbps

- A more suitable cycle rather than the globally unique cycle can been chosen, by considering the different conditions of stream characteristics, port bandwidth, topology etc. in different areas of the network

Summary

- Stream aggregation is beneficial in aspect of reducing the stream identification, per-stream state machines, as well as the per-hop buffer and delay for component streams after aggregation
- Buffer at aggregation/dis-aggregation points can be estimated if the scheduling mechanism is well designed
- The idea of variable cycles for aggregation is attractive, it is interesting to further investigate this idea, such the considering transformation between more cycles, potential combination with multilevel-CQF scheduling, etc.

Thank you.