Deadline based Forwarding

draft-peng-detnet-deadline-based-forwarding-09

Shaofu Peng	ZTE
Zongpeng Du	China Mobile
Kashinath Basu	Oxford Brookes University
Zuopin Cheng	New H3C
Dong Yang	Beijing Jiaotong University
Chang Liu	China Unicom

IETF-119 Mar 2024

Updates (07 -> 09)

- Clarify the process of the latency compensation under burst accumulation context according to the discussion in maillist.

- Supplement figure for each option for easy understanding.
  - Options with latency compensation is termed as CEDF, with the benefit of core stateless, and is recommended.

- Describe two implementation methods for on-time scheduling to absorb latency deviation E, and the related delay analysis.
  - E+D integration, or E|D decoupling

- Supplement common toplogy & service TSpec/RSpec examples.

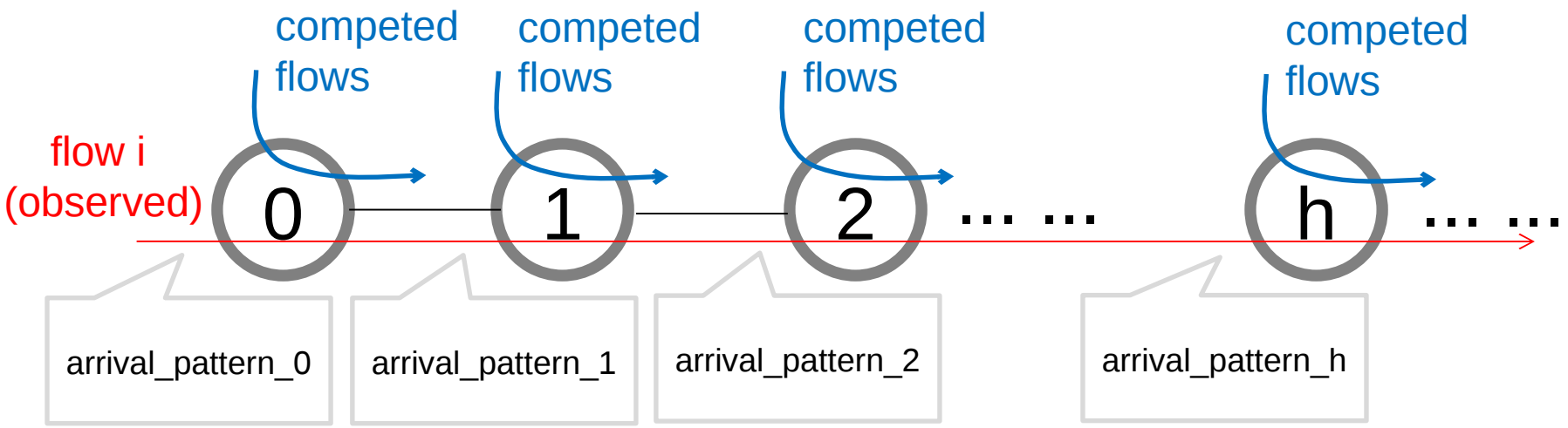
- Taxonomy considerations.

Motivations

- Challenges of the existing queueing mechanisms:
  - TSN ATS/CBS come with a high latency variance, as the minimum latency is not affected by them. The worst-case latency is **overestimated**, basically inversely proportional to the service rate. CBS can not even work independently, and should combine with re-shaping function (such as ATS) to avoid **burstiness cascade**.
  - TSN TAS requires time synchronization and has **scalability issues** on GCL calculation, update and installation.
  - TSN CQF requires time synchronization and relies on very small link delay. Although ECQF only requires frequence synchronization, but with **overprovision issues**.
  - The widely used priority based queuing scheme in IP/MPLS diff-serv network, may give better average latency, but with **worst bounded latency**.

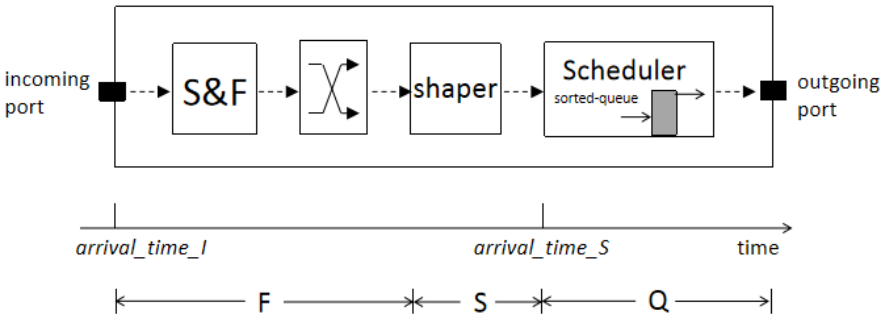
- To meet the large scaling requirements, this document **introduce EDF (Earliest Deadline Forwarding) scheduling to DetNet Data Plane**, to uniformly provide bounded delay/jitter by in-time/on-time mode.

Update 1: Clarification of Latency Compensation Processing



- For classical EDF, we aready have the conclusion that if the arrival pattern (or after regulation) of each flow meet its constraint funciton, EDF scheduler will work successfully.
- For latency compensation, with the help of latency deviation E, we can always get the ideal arrival pattern that is used by EDF scheduler to rank packets, no matter what is the shape of the real arrival pattern.
  - **ideal arrival_pattern_h = ideal arrival_pattern_0 + h*D**
  - **ideal arrival time = real arrival time + E**, e.g, for arrival_pattern_1, it is $t_0 + q + D - q = t_0 + D$, where t_0 is the ideal arrival time of pattern_0.

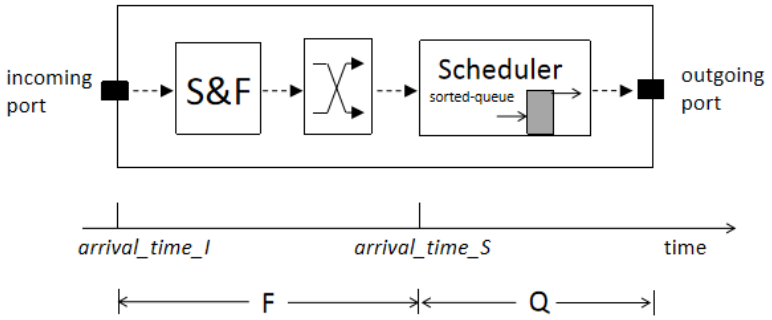
Update 2: 4 Options and their Internal Components



option-1

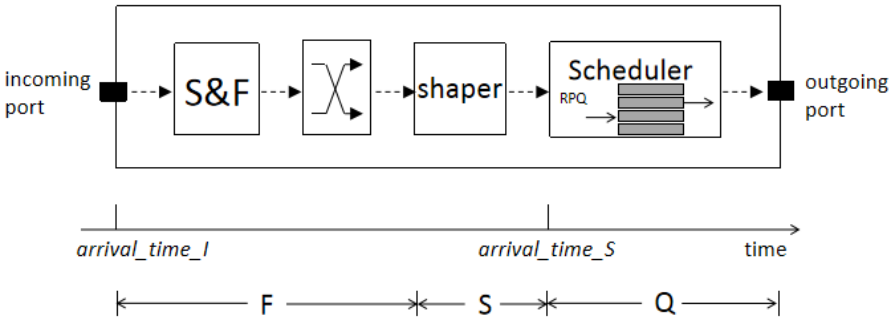
(rank = arrival_time_S + D - F)

CEDF:



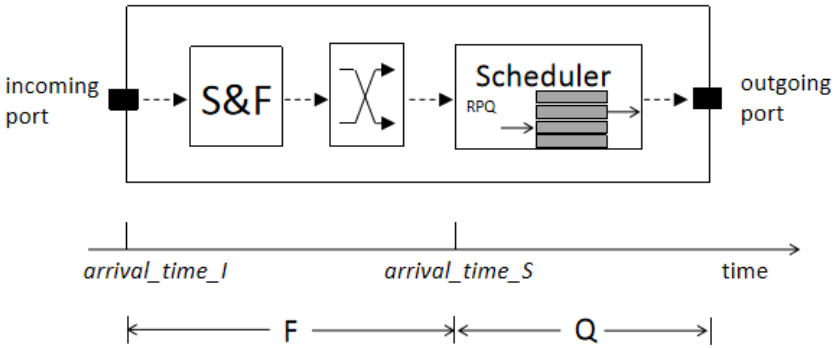
option-3

(rank = arrival_time_I + D + **E**)



option-2

(CT <= Q < CT+CTI, Q = D - F)



option-4

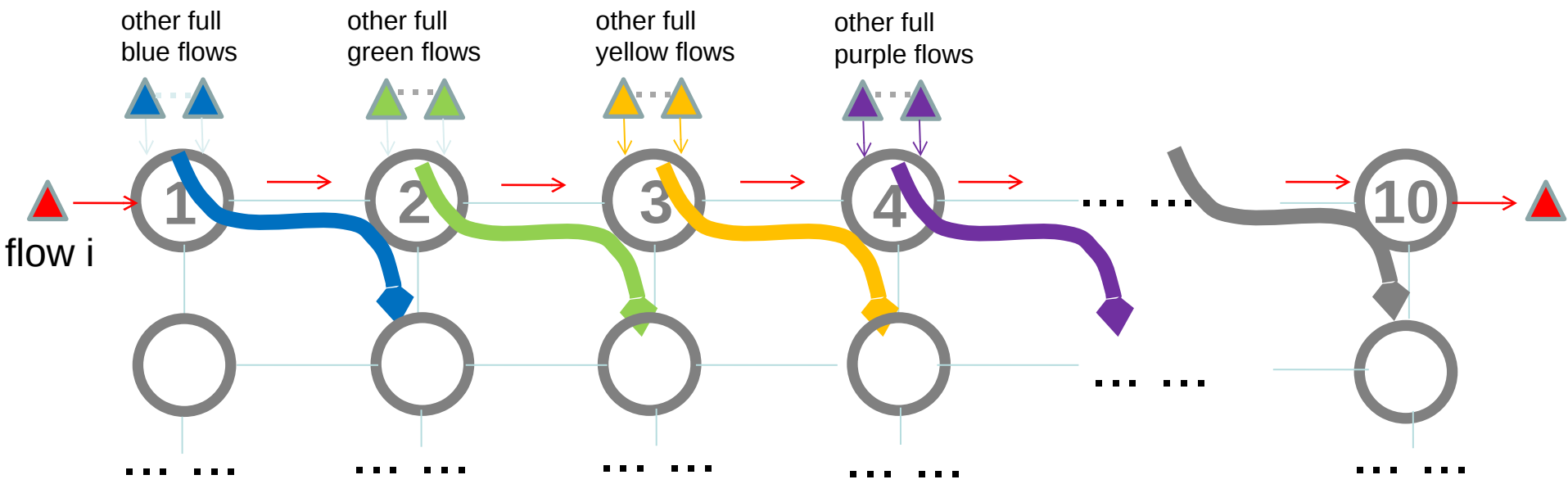
(CT <= Q < CT+CTI, Q = D + **E** - F)

Update 3: On-time Scheduling

- On-time mode can be further implemented in two methods:
  - **E+D integration**:
    - The packet is scheduled by the scheduler configured with on-time mode based on D+E.
    - The E2E latency is in the range [D*hops, D*hops+d_i].
  - **E|D decoupling**:
    - The packet is scheduled by pre-sheduler configured with on-time mode based on E, then scheduled by post-scheduler configured with in-time mode based on D.
    - The E2E latency is in the range [D*(hops-1), D*(hops-1)+d_i].
  - Both methods provide jitter of delay level value (d_i) if at the last hop the output fort faced to client also have the full competed flows, otherwise the jitter is just CTI.

Update 4: Common Topology Example

- Link speed: **100 Gbps**.
- flows passed through each interface:
  - **TSpec**: each flow has packet size **1000 bits,** average rate **10 Mbps**.
  - **RSpec: (not including link propagation delay)**
    - flow1~flow100 may tolerate E2E latency **100us**, and E2E jitter **10us or 100us**.
    - flow101~flow200 may tolerate E2E latency **200us**, and E2E jitter **20us or 200us**.
    - ... ...
    - flow901~flow1000 may tolerate E2E latency **1ms**, and E2E jitter **100us or 1ms**.
- Topogogy



- Resources for each delay level @ link (100 Gbps), supported number of flows (each with 10 Mbps).

Delay Levels	Resources	Admited Flows	E2E Delay & Jitter (10 hops)
d1 (10us)	b = 1000000 bits, r = 10 Gbps	1000	delay 100us, jitter 10us(on)/100us(in)
d2 (20us)	b = 900000 bits, r = 9 Gbps	900	delay 200us, jitter 20us(on)/200us(in)
d3 (30us)	b = 810000 bits, r = 8.1 Gbps	810	delay 300us, jitter 30us(on)/300us(in)
d4 (40us)	b = 729000 bits, r = 7.3 Gbps	729	delay 400us, jitter 40us(on)/400us(in)
d5 (50us)	b = 656100 bits, r = 6.6 Gbps	656	delay 500us, jitter 50us(on)/500us(in)
d6 (60us)	b = 590490 bits, r = 6.0 Gbps	590	delay 600us, jitter 60us(on)/600us(in)
d7 (70us)	b = 531450 bits, r = 5.3 Gbps	531	delay 700us, jitter 70us(on)/700us(in)
d8 (80us)	b = 478310 bits, r = 4.8 Gbps	478	delay 800us, jitter 80us(on)/800us(in)
d9 (90us)	b = 430480 bits, r = 4.3 Gbps	430	delay 900us, jitter 90us(on)/900us(in)
d10 (100us)	b = 387440 bits, r = 3.9 Gbps	387	delay 1ms, jitter 100us(on)/1ms(in)
		total: 6511	

1) **No overprovision**, i.e., addmision check is NOT based on burst size per each delay level duration.
2) **Customized per-hop latency**, i.e., per-hop latency of flow is based on its RSpec, NOT its TSpec (i.e., service flow rate).

Update 5: Taxonomy Considerations

- ***Per hop latency dominant factor***: Delay level itself.
- ***Non-periodic***: There is no defined periodic quantification unit of scheduling power.
- ***Asynchronous***: All DetNet flows arrive asynchronously, and schedulers work independently without synchronization (such as time synchronization).
- ***Class level***: DetNet Flows is grouped by delay levels.
- ***Work-conserving/non-work-conserving configurable***: Scheduler configured with in-time mode is work-conserving, while the EDF scheduler configured with on-time mode is non work-conserving.
- ***In-time/on-time configurable***: Scheduler may be configured with in-time mode or on-time mode.
- ***Delay based***: A DetNet flow is scheduled based on its expected delay level, rather on its reserved bandwidth (i.e., rate), to fit well the case of low bandwidth consumption but urgent flows.

Next step

- Any questions/comments ?

Thank you!