Latency Guarantee with Stateless Fair Queuing (C-SCORE)

draft-joung-detnet-stateless-fair-queuing-01

Jinoo Joung, Jeong-dong Ryoo, Tae-sik Cheung, Yizhou Li, Peng Liu

IETF 118, Nov. 08

Contents

- C-SCORE overview (3 pages, will be skipped.)
- Discussions about C-SCORE so far
  - Validation on Pay burst only once
  - Validation on E2E latency bound of C-SCORE
  - and more
- Updates on C-SCORE
  - Added texts
    1. Total reserved rate estimation with Metadata
    2. Clarification on Time difference compensation
  - Added Reference
    3. Scalable Flow Isolation with C-SCORE

Work Conserving Stateless Core Fair queuing (C-SCORE)

- Framework
  - FT, Finish time F(p) = Service order of packet p. Smaller FT gets earlier service.
  - At entrance node 0: $F_0(p) = \max\{F_0(p-1), A_0(p)\} + L(p)/r$;
  - At core node h: $F_h(p) = F_{h-1}(p) + d_{h-1}(p)$.
  - Whenever there are packets in the queue, the link never idles.
  - Packets in the queue are served in the ascending order of FT
- If $d_h(p) = Lmax_h/R_h + L/r$,
- Then the E2E latency of p’s flow is bounded [Kaur] by

$$\frac{B-L}{r} + \sum_{h=0}^{H}\left(\frac{Lmax_h}{R_h} + L/r\right)$$

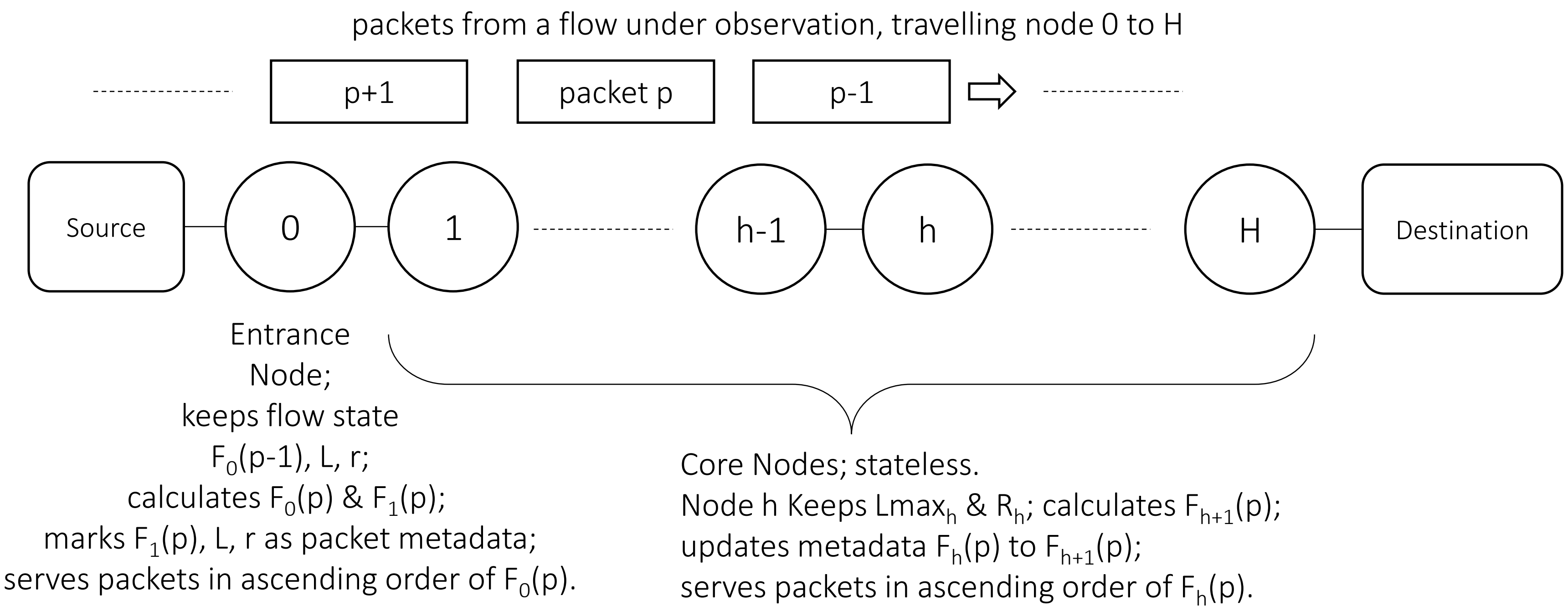
This bound is same with a stateful fair queuing network (PGPS, etc.)

$\frac{Lmax_h}{R_h}$ is the only term external & can be negligible.

- B, L, r are flow specific, which can be controlled according to requirement → Latency bound can be adjusted if necessary

Symbol	Definition
$F_h(p)$	‘Finish time’ of packet p at node h
$A_0(p)$	Arrival time of p at node 0
L(p)	Length of p
L	Max Packet Length of p’s flow
$d_h(p)$	‘Delay factor’ of packet p at h
ρ_j	Arrival rate of flow j
Bj	Max burst of flow j
r, r(p)	Service rate of p’s flow
$r_{h,j}$	Service rate of flow j at node h
$Lmax_h$	Max Packet Length at node h
R_h	Link capacity of h
f(h)	Set of flows in node h

C-SCORE Framework Overview


packets from a flow under observation, travelling node 0 to H
p+1
packet p
p-1
Source
0
1
h-1
h
H
Destination

Entrance Node; keeps flow state $F_0(p\text{-}1)$, L, r; calculates $F_0(p)$ & $F_1(p)$; marks $F_1(p)$, L, r as packet metadata; serves packets in ascending order of $F_0(p)$.

Core Nodes; stateless.
Node h Keeps $Lmax_h$ & R_h; calculates $F_{h+1}(p)$;
updates metadata $F_h(p)$ to $F_{h+1}(p)$;
serves packets in ascending order of $F_h(p)$.

Symbol	Definition
$F_h(p)$	'Finish time' of packet p at node h
$A_0(p)$	Arrival time of p at node 0
L(p)	Length of p
L	Max Packet Length of p's flow
$d_h(p)$	'Delay factor' of packet p at h
ρ_j	Arrival rate of flow j
Bj	Max burst of flow j
r, r(p)	Service rate of p's flow
$r_{h,j}$	Service rate of flow j at node h
$Lmax_h$	Max Packet Length at node h
R_h	Link capacity of h
f(h)	Set of flows in node h

C-SCORE Operational procedures

1. Network configuration stage
   - A source requests latency bound for flow i, with specifying its ρ_i and Bi
   - If the latency bound can be met, admit the flow
   - Network reserves the links in the path such that
     - $\rho_j \leq r_{h,j}$ and $\sum_{j \in f(h)} r_{h,j} \leq R_h$, for all h
2. The entrance node or the source
   - Maintains the flow state, i.e. $F_0(p-1)$ & r.
   - Maintains a clock, for $A_0(p)$.
   - Maintains the link info $Lmax_0/R_0$.
   - Upon receiving or generating packet p,
     - Obtains $F_0(p) = \max\{F_0(p-1), A_0(p)\} + L(p)/r$. Use it as the FT in 0. Put p in a sorted queue.
     - Obtains $F_1(p) = F_0(p) + Lmax_0/R_0 + L/r$.
     - Records $F_1(p)$ & L/r in the packet as metadata for the use in the next node 1.
   - Update the flow state to $F_0(p)$.
3. A core node h
   - Maintains the link info $Lmax_h/R_h$. (A rather static value)
   - Upon receiving packet p,
     - retrieve meta-data $F_h(p)$ & L/r, use $F_h(p)$ as the FT. Put p in a sorted queue.
     - Obtain $F_{h+1}(p) = F_h(p) + Lmax_h/R_h + L/r$.
     - Update metadata $F_h(p)$ with $F_{h+1}(p)$ before or during p is in the queue.

Symbol	Definition
$F_h(p)$	'Finish time' of packet p at node h
$A_0(p)$	Arrival time of p at node 0
L(p)	Length of p
L	Max Packet Length of p's flow
$d_h(p)$	'Delay factor' of packet p at h
ρ_j	Arrival rate of flow j
Bj	Max burst of flow j
r, r(p)	Service rate of p's flow
$r_{h,j}$	Service rate of flow j at node h
$Lmax_h$	Max Packet Length at node h
R_h	Link capacity of h
f(h)	Set of flows in node h

Discussions about C-SCORE so far

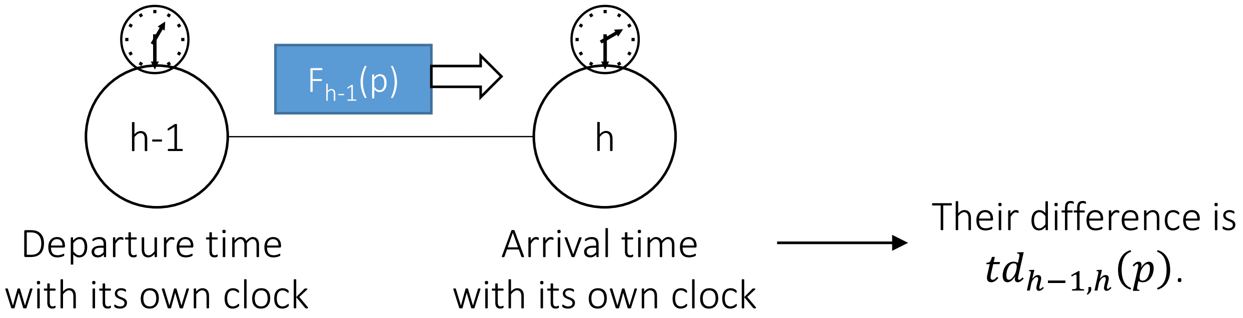
- Validation of “Pay burst only once” property
  - which is enjoyed by rate-based queuing schemes, e.g. ATS, DRR, FQ, C-SCORE
  - e.g. C-SCORE’s E2E latency bound $\frac{B-L}{r} + \sum_{h=0}^{H}(\frac{Lmax_h}{R_h} + L/r)$ has the term (B-L)/r that is independent of hop count.
  - However, the latency bound in a single hop is (B-L)/r + $Lmax_h/R_h$ + L/r.
  - Thus, sum of {bound in each hop} > E2E latency bound
  - This nice property does not apply to per-hop latency guaranteeing approaches.
- Validation of C-SCORE’s E2E latency bound
  - based on the original paper [Kaur].
  - with a specific network example.
- Effectiveness of Time difference compensation
- Finish Time (FT)’s range, precision, and required bits

1: Total reserved rate estimation with Metadata

- Recent updates on the Scaling Requirement draft:
    - “This resource allocation complexity does not directly affect achievable end-to-end latency and jitter bounds, but it does surface in other areas such as the amount of computation and elapsed time required to admit a new flow to a DetNet network without disrupting the DetNet QoS being provided to already admitted flows.”
- Any queueing scheme requires an admission control to ensure that the sum of the reservation rates of all flows that traverse any link in the network is no larger than the link capacity.
- There can be partial reservation failure, signaling packet loss, or node failure. Thus, a node needs to “remember” what decision it made for the flow in past → Flow state management is required. (e.g. RSVP maintains “soft-state”.)
- Estimating the current total reserved rate without per-flow state would help simplifying and stabilizing the distributed signaling system.
- Added: “The metadata carried for C-SCORE in packets can be used for estimating the total reserved service rate in a core node, as in the following.” (See Appendix.)

2: Considerations of Time difference between nodes

- In reality, there are time differences between nodes, including the discrepancies of clocks and differences due to the propagation delays.



Departure time with its own clock

Arrival time with its own clock

Their difference is $td_{h-1,h}(p)$.

- Note that FT does not need to be precise. It is used just to indicate the packet service order. Therefore, we can assume that the propagation delay is constant and the clocks do not drift.
- $td_{h-1,h}(p)$ can be simplified to a constant value, $td_{h-1,h}$.
- In this case the delay factor should be modified to be

$$d_h(p) = \frac{Lmax_h}{R_h} + L/r + td_{h,h+1}.$$

- The E2E latency bound increases as much as the sum of propagation delays from node 0 to h.
- C-SCORE does not need global time synchronization.

Based on Andrea’s comments, the following sentences are added:

> “By the time difference compensation, the nodes become aware of the global clock discrepancies using a periodic quantification of the local clock discrepancies between adjacent nodes. Link by link, this ends up producing awareness of the discrepancies between the clocks of the ingress nodes, which is then included in the computation of the FTs in core nodes. It is not synchronization in a strict sense because it does not involve the re-alignment of the clocks, only the quantification of their differences.”

3: Added reference [C-SCORE]

- Joung J., Kwon J., Ryoo J-D., Cheung T., "Scalable flow isolation with work conserving stateless core fair queuing for deterministic networking" IEEE Access, vol. 11, Sep. 2023. doi: 0.1109/ACCESS.2023.3318479
- Validates C-SCORE with extensive simulations
  - Topology I: 9 nodes, 36 flows, 7 hops network
  - Topology II: 80 nodes, ~300 flows, 16 hops network
  - Various Delay factor functions
  - Implemented with PIFO; or with FIFO of similar flows
  - Comparisons with ATS, FIFO, DRR, Stateful Virtual Clock (VC)
  - Comparison with non-work conserving core stateless fair queuing [Stoica]
- In every setup, C-SCORE
  - performs almost the same with stateful VC.
  - is much superior to ATS, FIFO, DRR, and non-work conserving stateless FQ.
  - meets the theoretical E2E latency bound.

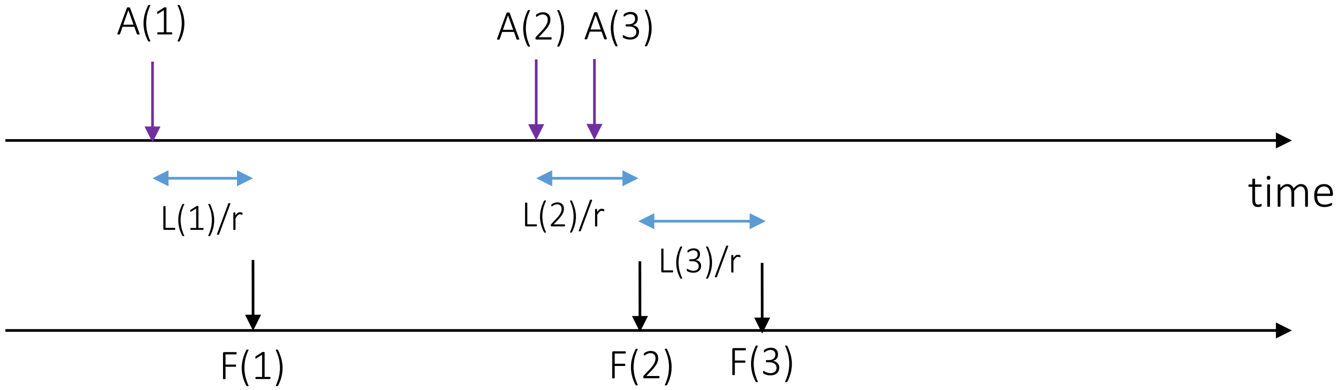
Appendix

Estimating the total reserved service rate in a link

Mitigating Resource allocation complexity by Estimation of total reserved service rate at a link

$F_0(p) = \max\{F_0(p-1), A_0(p)\} + L(p)/r; F(0)=0.$

Single flow case:



Mitigating Resource allocation complexity by Estimation of total reserved service rate at a link

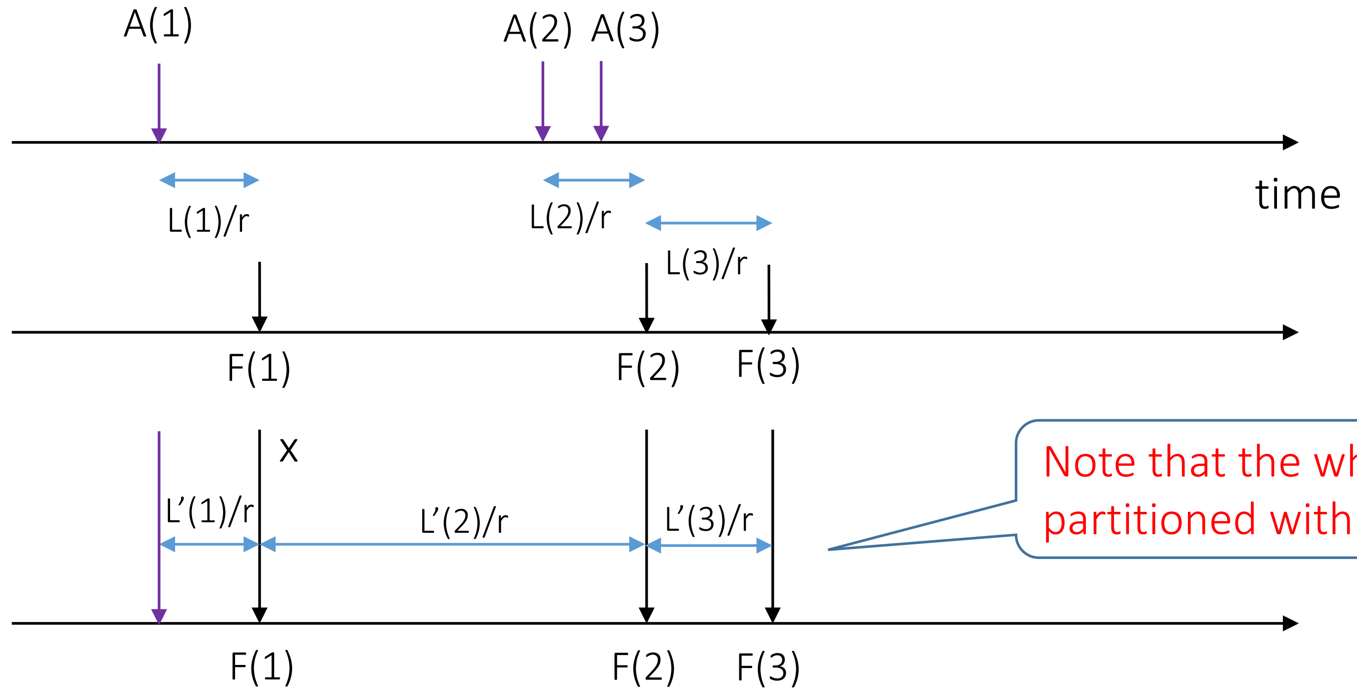
$F_0(p) = \max\{F_0(p-1), A_0(p)\} + L(p)/r$; $F(0)=0$.

Let $\mathbf{F_0(p) = F_0(p-1) + L'(p)/r}$.

Since $F_h(p) = F_{h-1}(p) + d_{h-1}(p)$; and $d_h(p)$ is a function of node & flow only;

$\mathbf{F(p) = F(p-1) + L'(p)/r}$ **for any node.**

Single flow case:



Note that the whole time is partitioned with (F(p-1), F(p)].

Mitigating Resource allocation complexity by Estimation of total reserved service rate at a link

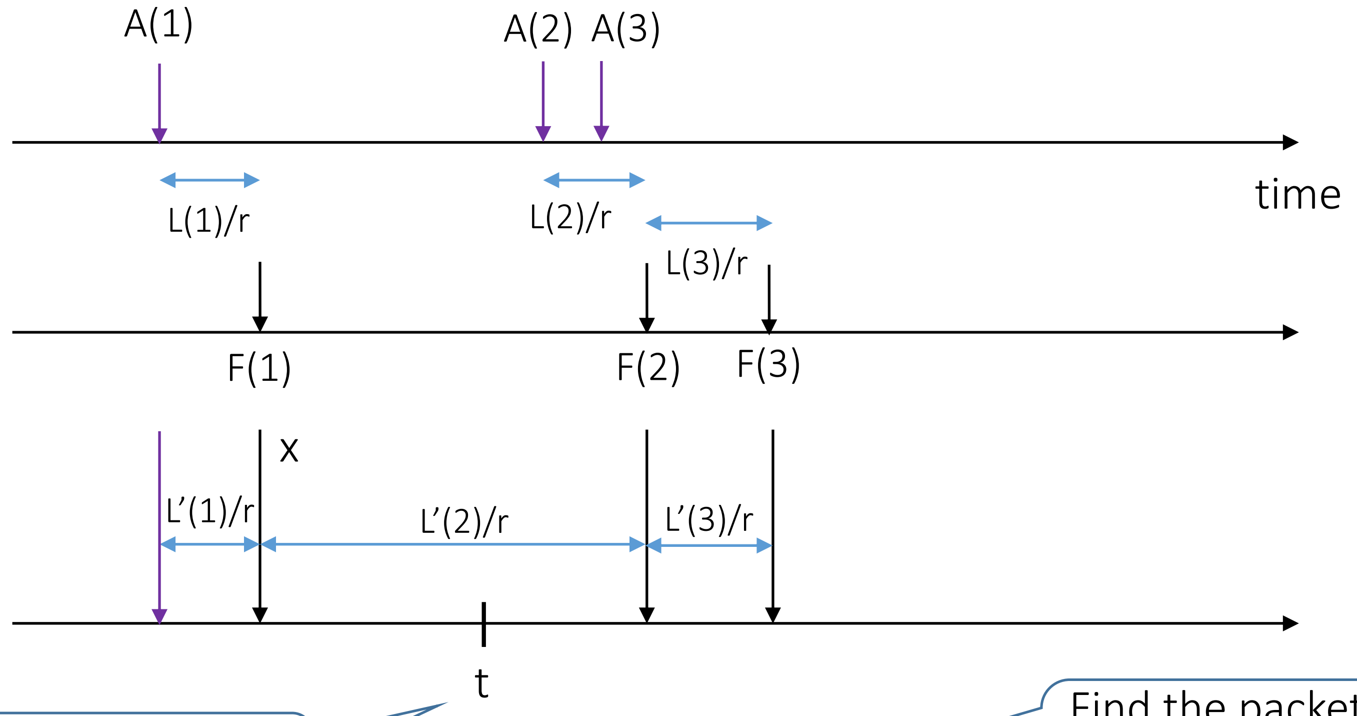
$F_0(p) = \max\{F_0(p-1), A_0(p)\} + L(p)/r$; $F(0)=0$.

Let **$F_0(p) = F_0(p-1) + L'(p)/r$**.

Since $F_h(p) = F_{h-1}(p) + d_{h-1}(p)$ and $d_h(p)$ is a function of node & flow;

$F(p) = F(p-1) + L'(p)/r$ for any node.

Single flow case:



We want to know the service rate of the flow at time t.

$F(2)-L'(2)/r < t < F(2)$

Find the packets that satisfies $F(p)-L'(p)/r < t < F(p)$. Read the metadata r written in this packets.

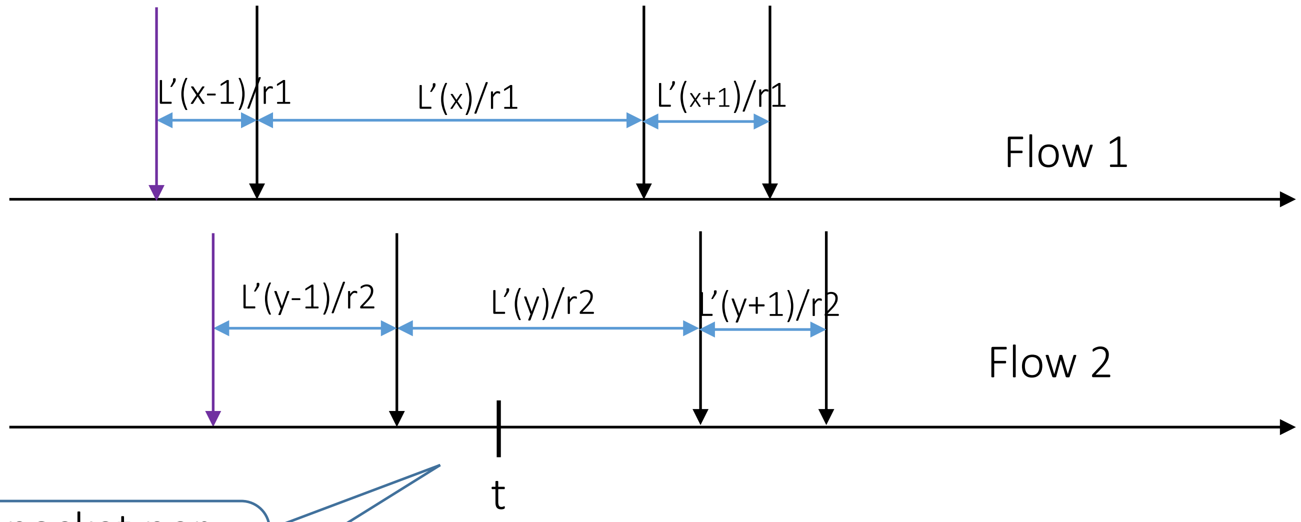
Estimation of total reserved service rate at a link: Multiple flows case

- Similarly, we can estimate r1+r2 by,

1) finding all the packets that satisfies $F(p)-L'(p)/r(p) < t < F(p)$.
2) summing the metadata $r(p)$ written in these packets.

- We need extra metadata $L'(p)$ and $r(p)$.
- We can also find the number of active flows and their service rates

Multiple flows case:



There is one & only one packet per flow whose FT "encompasses" any time t.

$F(x)-L'(x)/r1 < t < F(x)$ &
$F(y) -L'(y)/r2 < t < F(y)$

Thank you

- Please take a look at

  https://datatracker.ietf.org/doc/draft-joung-detnet-stateless-fair-queuing/

- Comments and Questions are welcome!

- [Bhagwan00] Ranjita Bhagwan and Bill Lin, “Fast and Scalable Priority Queue Architecture for High-Speed Network Switches", IEEE Infocom 2000 Conference, 26-30 March 2000
- [Sivaraman16] Anirudh Sivaraman, et. al. “Programmable Packet Scheduling at Line Rate”, ACM SIGCOMM ’16, August 22 - 26, 2016
- [Kaur] Jasleen Kaur, and Harrick M. Vin. "Core-stateless guaranteed rate scheduling algorithms." In Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213), vol. 3, pp. 1484-1492. IEEE, 2001.
- [ADN] Jinoo Joung, Juhyeok Kwon, Jeong-Dong Ryoo, and Taesik Cheung. "Asynchronous Deterministic Network Based on the DiffServ Architecture." IEEE Access 10 (2022).
- [Zhang] Lixia Zhang. "Virtual clock: A new traffic control algorithm for packet switching networks." In *Proceedings of the ACM symposium on Communications architectures & protocols*, pp. 19-29. 1990.
- [Stoica] Ion Stoica and Hui Zhang. "Providing guaranteed services without per flow management." *ACM SIGCOMM Computer Communication Review* 29, no. 4 (1999): 81-94.
- [Stiliadis] Dimitrios Stiliadis and Varma Anujan. "Rate-proportional servers: A design methodology for fair queueing algorithms." IEEE/ACM Transactions on networking 6, no. 2 (1998): 164-174.