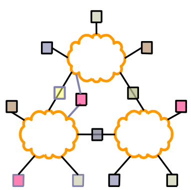
Packet Scheduling Algorithms

Nguyen Van Hung

Email: hungnv.vnu.uet@gmail.com

July, 20th 2017



Outline

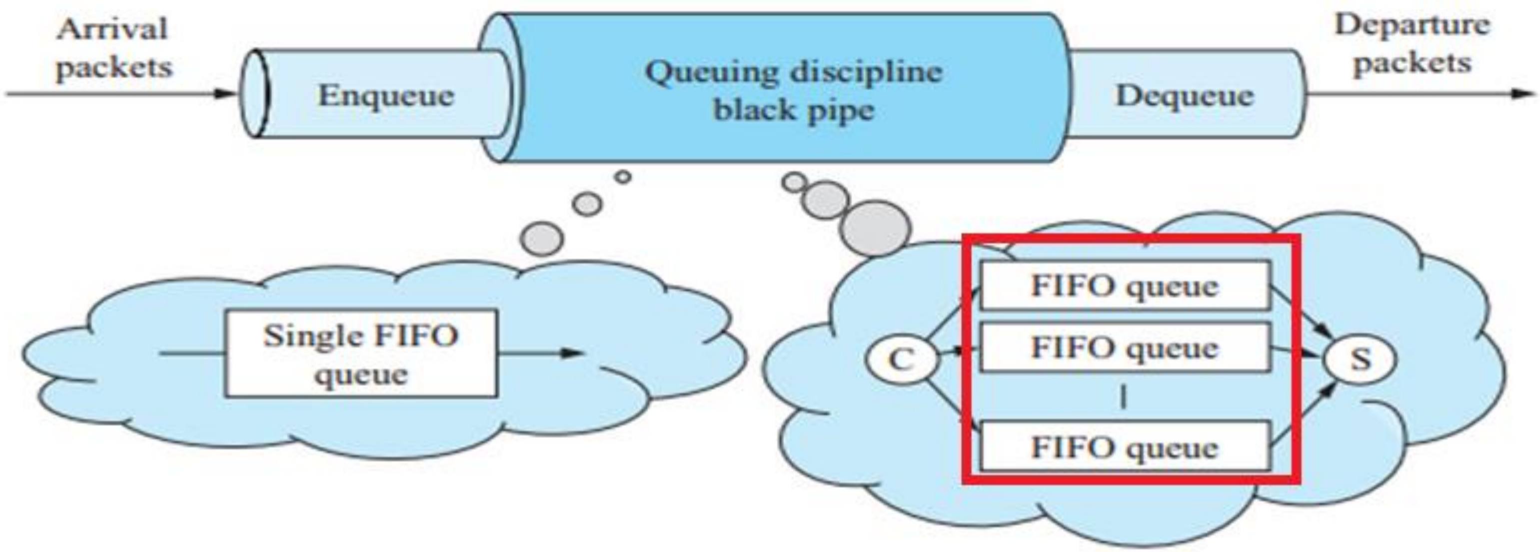
1) Motivation
2) Related Works
3) Approach
4) Schedule
5) References

Outline

1) Motivation
2) Related Works
3) Approach
4) Schedule
5) References

Motivation

Concept and possible architectures of scheduling (*)



- To enforce resource sharing between different flows.
- To provide an exact guarantee of fair sharing.

(*) Lin, Ying-Dar. *Computer networks: an open source approach*. McGraw-Hill, 2012

Outline

1) Introduction
2) Related Works
3) Approach
4) Schedule
5) References

Related Works

Scheduling algorithms:

- Based on packet sizes and flow's weighted to decide the number packets out.
  - ✓ For example: Deficit Round Robin [Shreedhar et al.,1996].

- Computed virtual finish time based on the allocated bandwidth and the arrival time of packets.
  - ✓ For example: Worst case Fair Weighted Fair Queuing [Bennett et al., 1996]

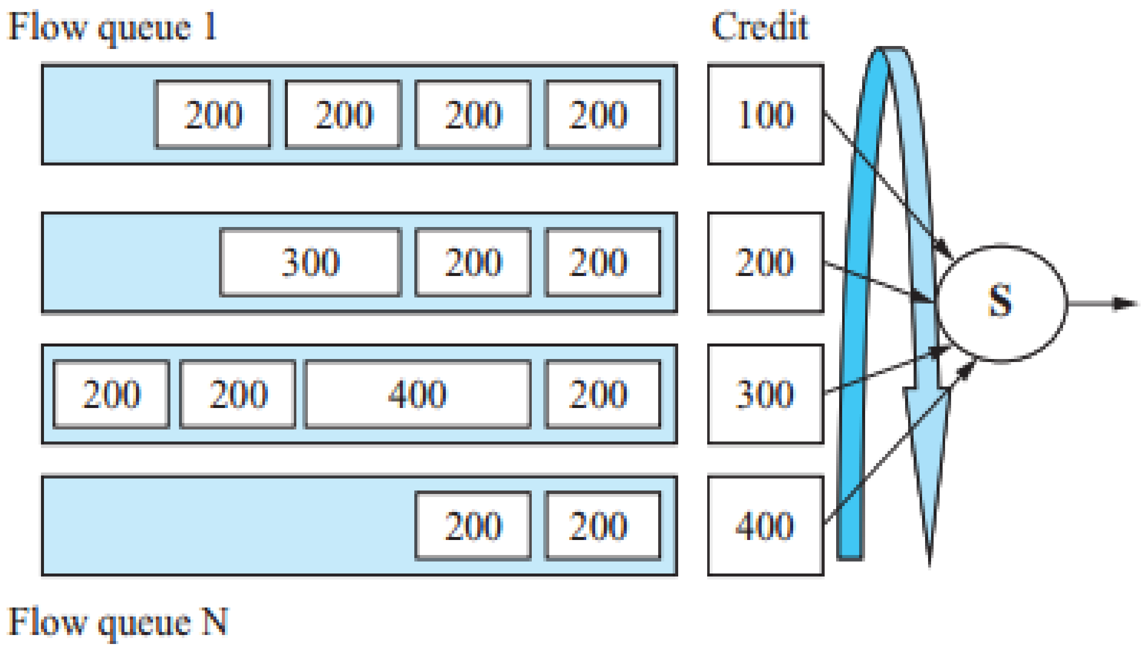
Popular Algorithms

- Fair Queuing is the most famous approach: guarantee worst case delay bound and fairly share resource

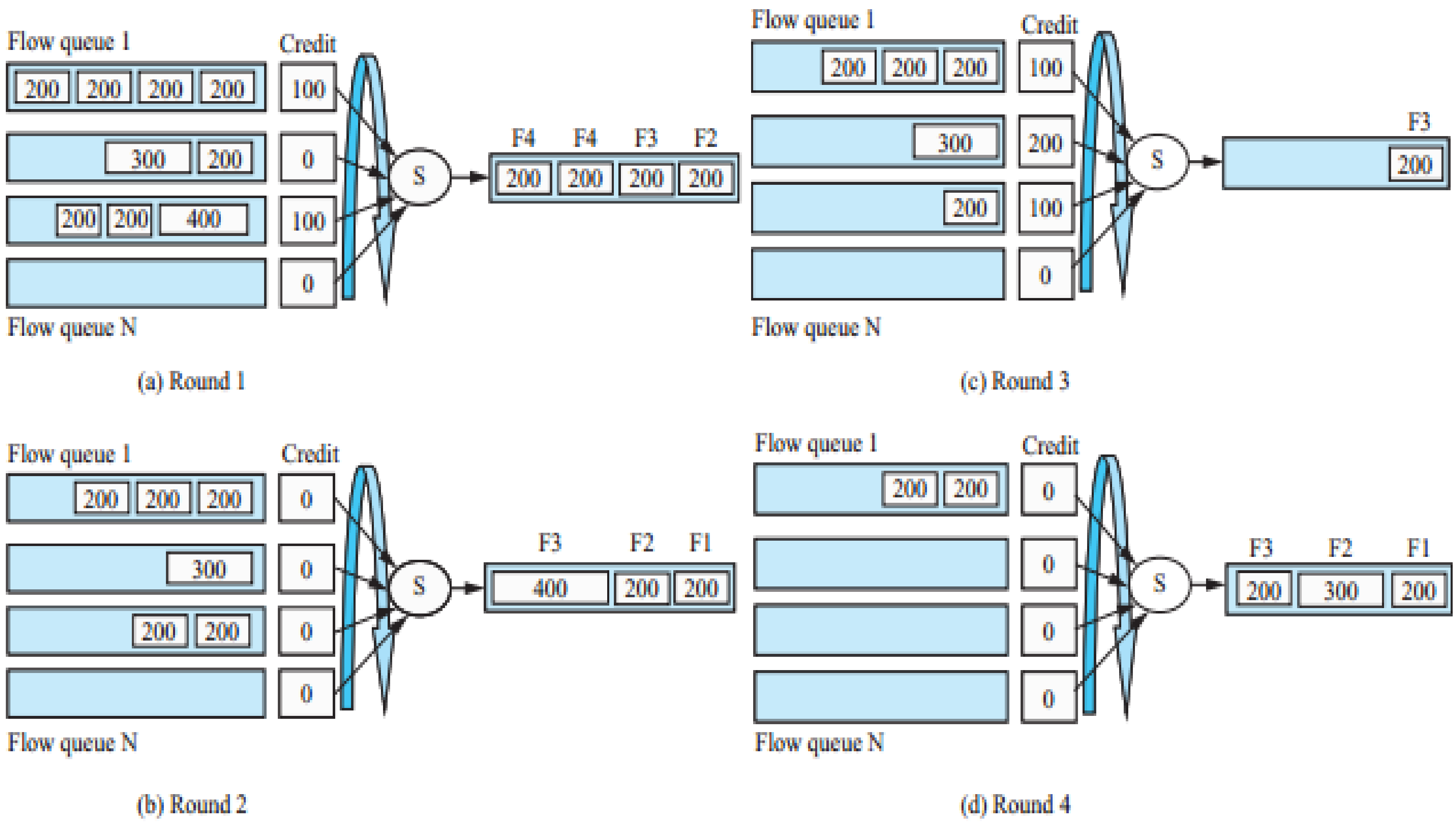
- Two popular algorithms:
  1. Round robin based: Deficit Round Robin (DRR)
     - Complexity: O(1)
  2. Sorted base: Weighted Fair Queuing (WFQ)
     - Complexity: O(log(n))

Deficit Round Robin: idea

- DRR limits each flow by the number of bytes sent in one round.
- A deficit counter is maintained for each flow to keep track of the allowed amount of data in this round.



Deficit Round Robin: examples

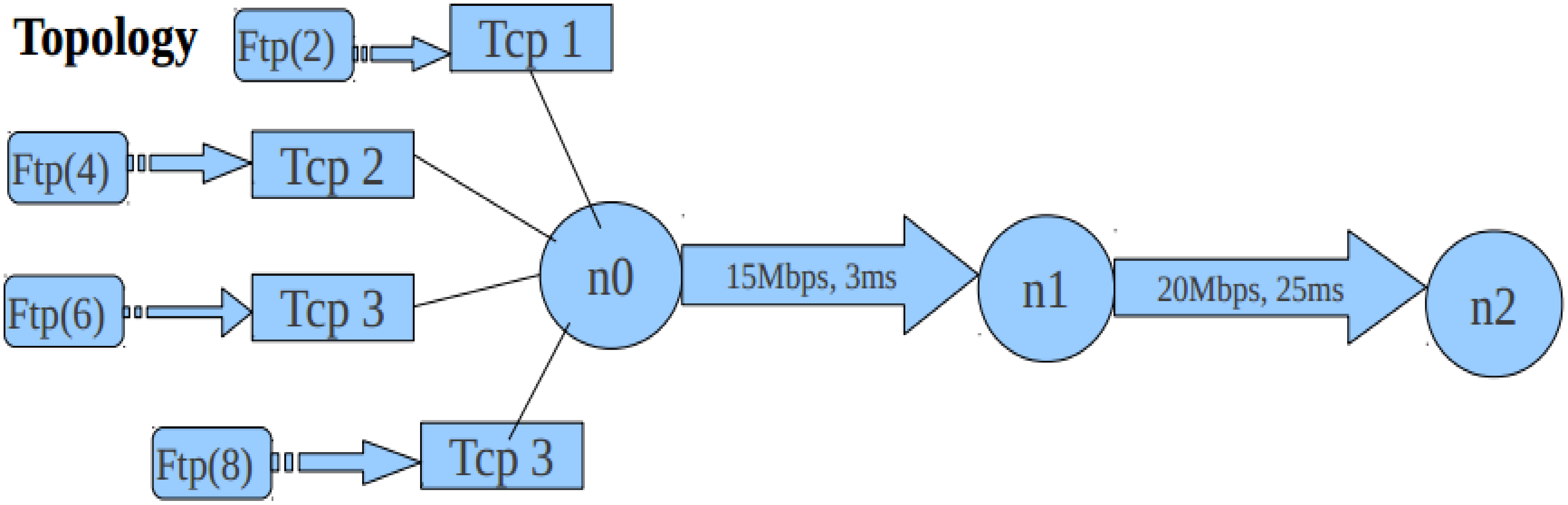


Deficit Round Robin: properties

- Advantages
  - It is simple to implement DRR algorithm (O(1))
- Disadvantages
  - Each flow gets the desired bandwidth over a long timescale.
  - A flow might wait a long time to send out packets if the number of flows becomes large.
  - Some packets might be sent out quickly and others might be served slowly, depending on their arrival time to the queue.

Simulation

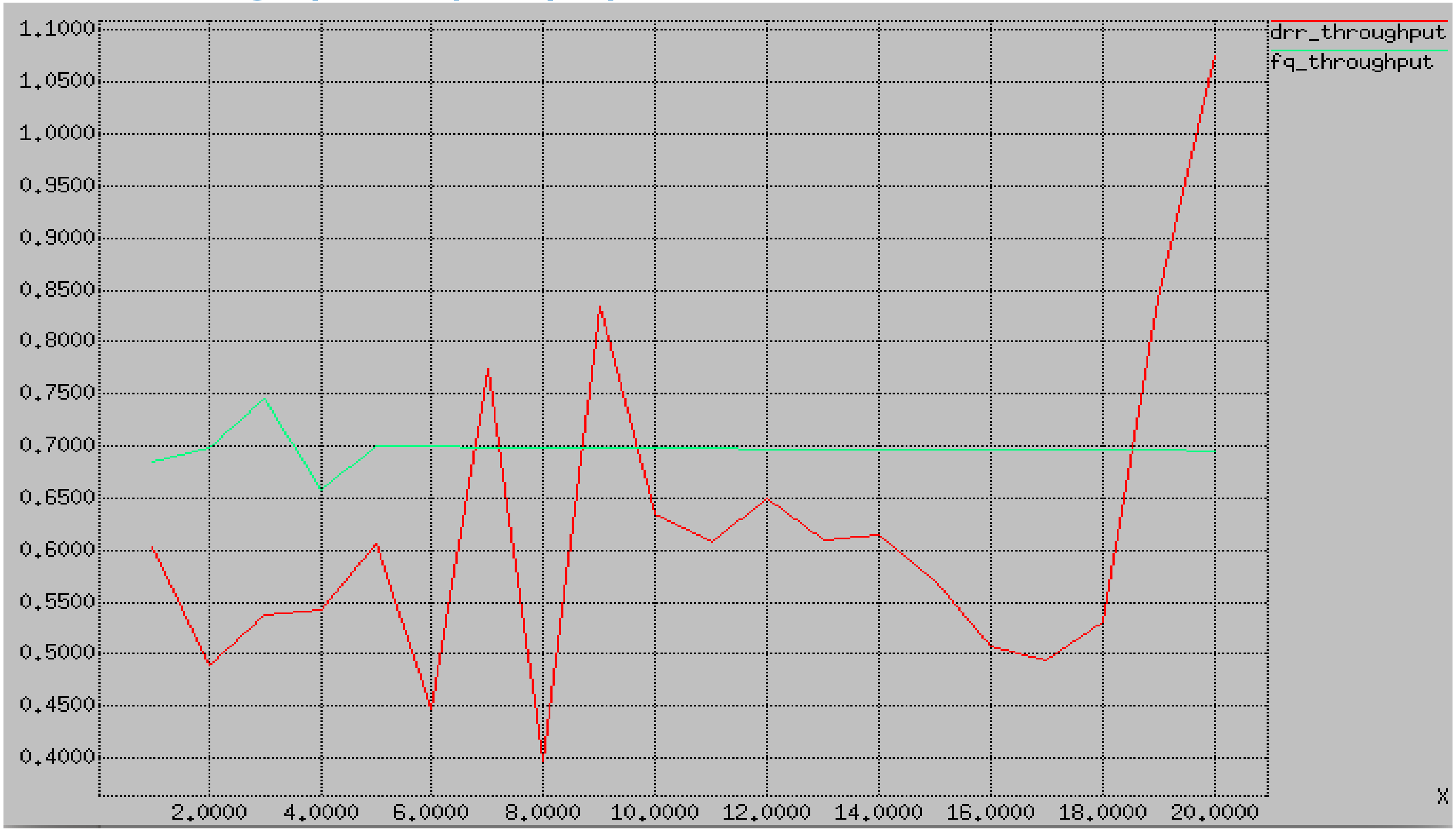
- Topology (*)



(*)https://github.com/vsubhashini/queue-ns2/tree/master/2-DRRvsFQ

Deficit Round Robin vs Fair Queuing

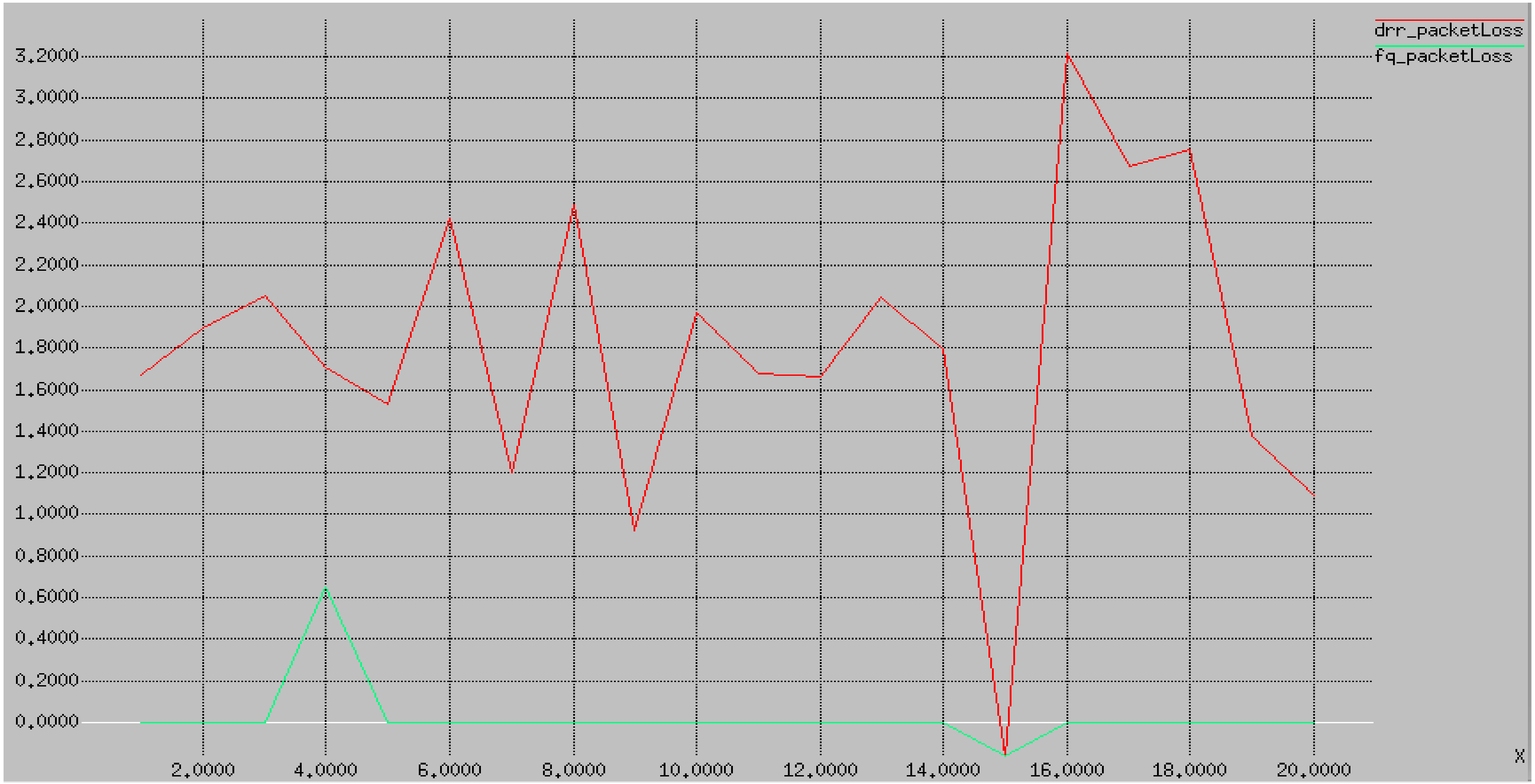
- Throughputs (Mbps)



Flow Id

Deficit Round Robin vs Fair Queuing

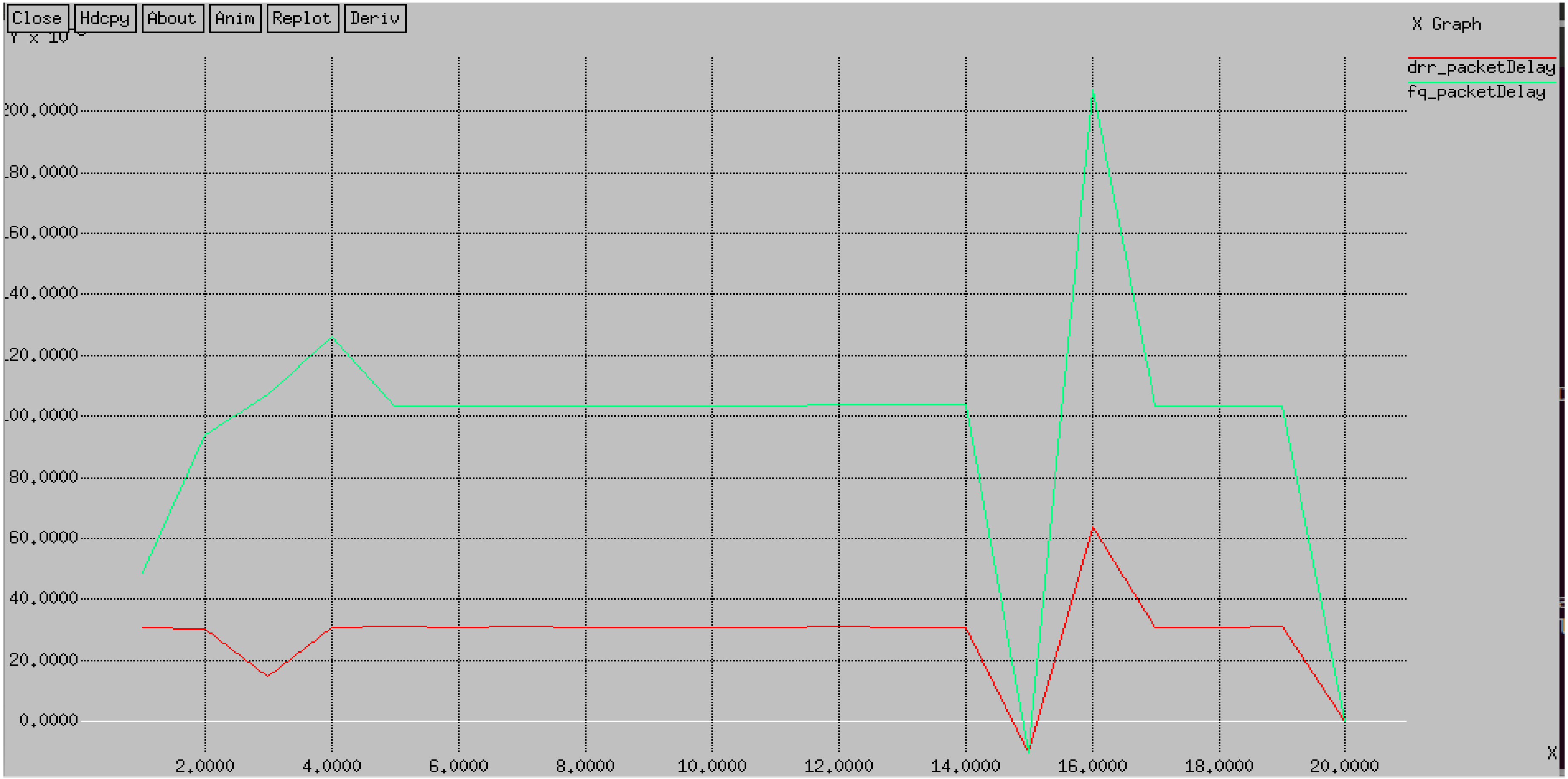
- Packet Loss (%)



Flow Id

Deficit Round Robin vs Fair Queuing

- Packet Delay (ms)



Flow Id

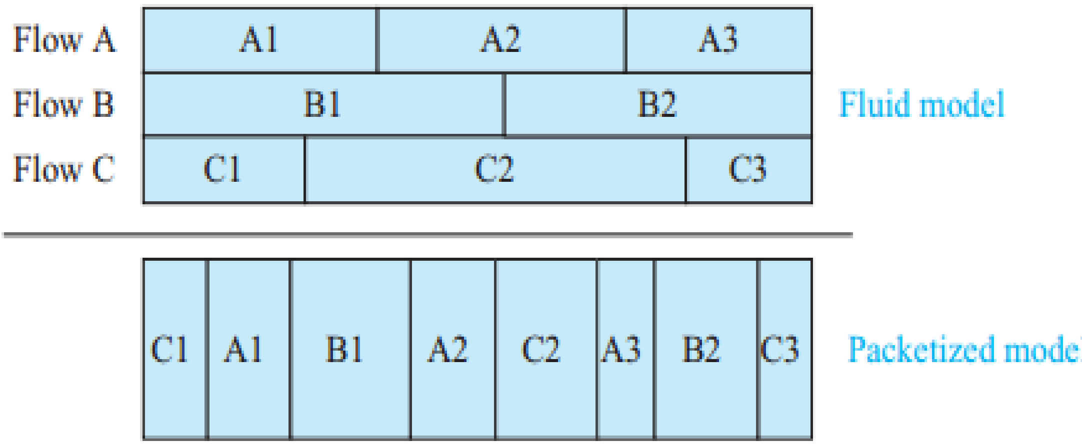
Deficit Round Robin vs Fair Queuing

- Getting result of simulation:
  - DRR throughput are less than FQ
  - FQ has a larger average delay than DRR
  - Packet loss of DRR is greater than FQ

- From these results we can say that
  - FQ provides advantages (compared with DRR) such as a fair resource allocation of bandwidth
  - However its corresponding delay percentage is higher

Weighted Fair Queuing : idea

- Fluid model Vs. Packetized model



- Packet transmissions from different flows proceed simultaneously in the fluid model but are interleaved in the packetized model
- packetized GPS (PGPS) – Weighted Fair Queuing: we can calculate the order in which the packets are sent out in the fluid model based on the size and arrival time of packets

Weighted Fair Queuing : idea

- Virtual time:

$$V(0) = 0$$

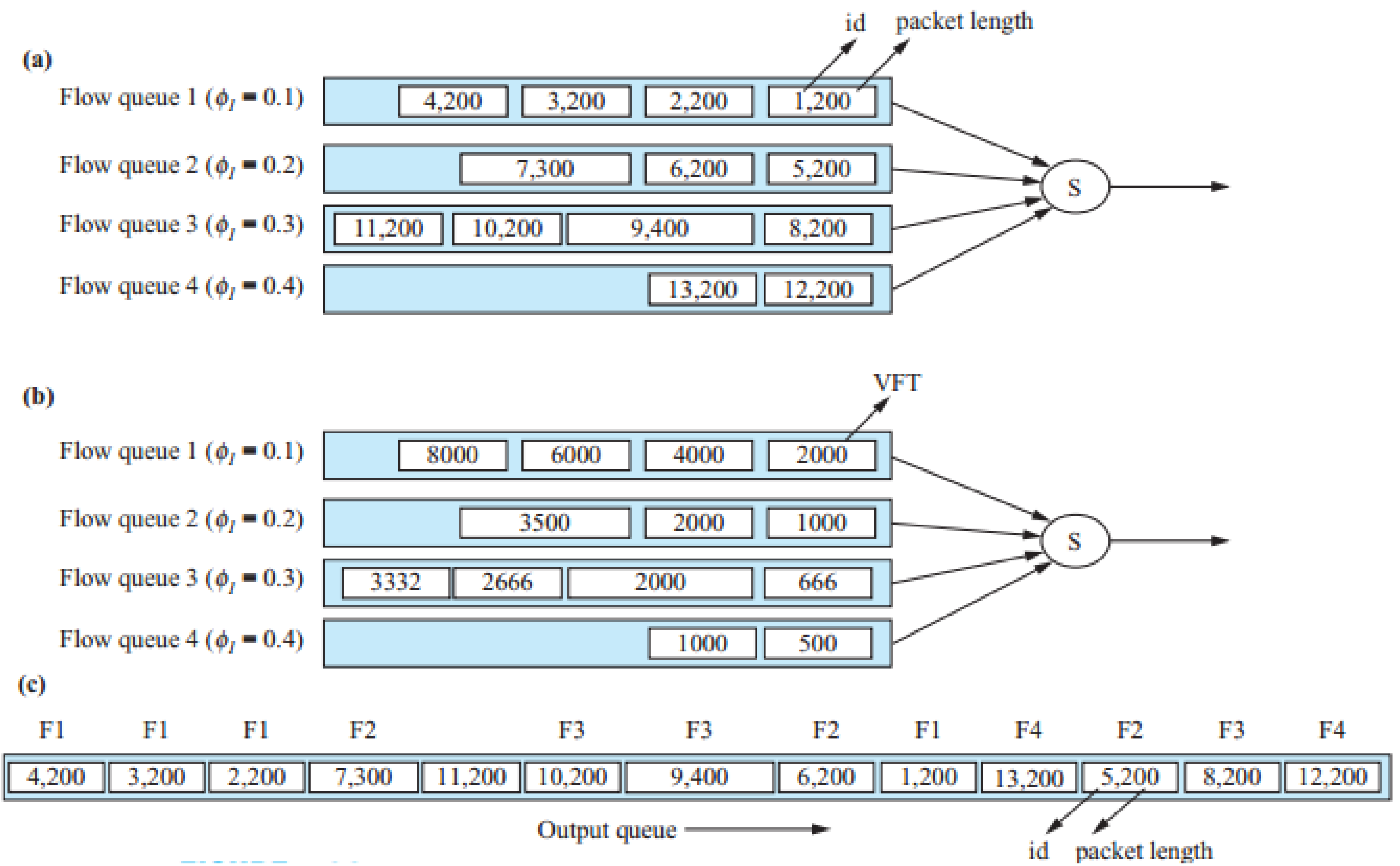
$$V(t_{j-1} + \tau) = V(t_{j-1}) + \frac{\tau}{\sum_{i \in B_j} \phi_i}$$

$$\tau \le t_j - t_{j-1},\ j = 2, 3, \ldots$$

$$F_k^0 = 0 \quad \Rightarrow \quad \begin{aligned} S_i^k &= \max\{F_i^{k-1}, V(a_i^k)\} \\ F_i^k &= S_i^k + \frac{L_i^k}{\phi_i} \end{aligned}$$

Weighted Fair Queuing: examples

- Assume all packets arrive at the same time, V(t) = 0



Weighted Fair Queuing: properties

- Advantages:
    1. The virtual time finishing times can be determined at the packet arrival time.
    2. The packets are served in order of virtual time finishing time.
    3. We need only update virtual time when there are events in the GPS system.

- Disadvantages:
    - The price to be paid for these advantages is some overhead in keeping track of backlogged sessions which is essential in the updating of virtual time.

Outline

1) Introduction
2) Related Works
3) Approach
4) Schedule
5) References

Main tasks

- Current formula to compute virtual finishing time:

$$F_i^k = F_i^{k-1} + \frac{L_i^k}{\phi_i}$$

- Construct a new formula to compute virtual finish time based on capacity:

1. Investigate the implementation of Weighted Fair Queuing
2. Proposed a new formula to compute virtual finish time based on capacity
3. Proved the delay bound and fairness of new algorithm
4. Implement new algorithm and check the result
5. If result is OK, start writing paper

Difficulties

- Analytical Result: Proved
  - Delay bound
  - Fairness

Outline

1) Introduction
2) Related Works
3) Approach
4) Schedule
5) References

Schedule: 8/2017–7/2018 (12 months)

Tasks	Time
Investigate the implementation of Weighted Fair Queuing	Month: 1
Propose a new formula to compute virtual finish time based on capacity	Month: 2- 4
Prove the delay bound and fairness of new algorithm	Month: 5-7
Implement the new algorithm and check the experimental results	Month: 8-10
Write a conference/journal paper	Month: 11- 12

Outline

1) Introduction
2) Related Works
3) Approach
4) Schedule
5) References

References

1. Lin, Ying-Dar. *Computer networks: an open source approach*. McGraw-Hill, 2012.
2. Parekh, Abhay K., and Robert G. Gallager. "A generalized processor sharing approach to flow control in integrated services networks: the single-node case." *IEEE/ACM transactions on networking* 1.3 (1993): 344-357.
3. Bennett, Jon CR, and Hui Zhang. "WF/sup 2/Q: worst-case fair weighted fair queueing." *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*. Vol. 1. IEEE, 1996.
4. Shreedhar, Madhavapeddi, and George Varghese. "Efficient fair queuing using deficit round-robin." *IEEE/ACM Transactions on networking* 4.3 (1996): 375-385.