CS 7260 Internet Arch. Protocols Scribe Notes

Name: Gaurav A. Kanitkar

Date: 3rd Oct 2012

• Packet Scheduling

Packet scheduling refers to the process that decides the order in which the packets need to be processed so as to have an optimal throughput. It is quite similar to process scheduling in Operating Systems context.

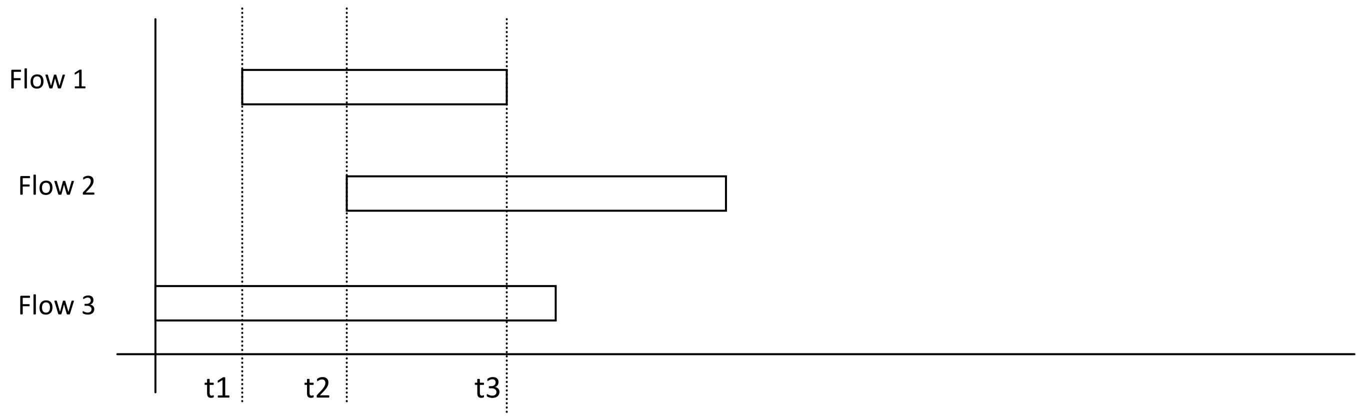
- **Main Purposes of Packet Scheduling:**
  1. <u>To provide bandwidth guarantees to flows (TCP flow / UDP flow):</u>
     E.g. For watching a video, we require a higher bandwidth whereas for VoIP calls much lower b/w is required. The network should guarantee the required bandwidth in each case.

  2. <u>To Provide Delay guarantees:</u>
     E.g. In real time systems (like VoIP calls again), we require a feedback without any delay. The network must be able to satisfy this requirement.

We now take a look at an Optimal Packet Scheduling policy called as "Generalized Processor Sharing" (GPS)

Generalized Processor Sharing:

This basically refers to the idea of providing CPU time slices to different processes and is on the same lines as the round robin policy carried out in Operating System process scheduling. We can also term this as an Infinitesimal round robin meaning that each packet gets service for some time in case of multiple packets.



Assume that rate = 1bit/s
Similar to OS, the packets get an equal share of time and the service keeps on rotating from one packet to another.

Let us consider an example just for the sake of understanding.
In the diagram drawn above,

T1: From the start till packet from flow 1 arrives, the packet from Flow 3 is the only one receiving service. For that time, the packet will receive full rate i.e. 1 bit/second.

T2: Packets from Flows 1 and 3 will receive service at rate 0.5 bits/s

T3: Packets from each flow will receive service at the rate of 1/3 bits per second.

So, this brings up the question as to why we don't use this 'Idealized' scheme in real life??
We don't use this scheme in real life because it does not make sense to transmit packets bit by bit. Instead a packet is always transmitted from start to end. We can always use this scheduler as a reference model for other schedulers.

- Let's take a look at a discrete version of a *fluid scheduler*: (Fluid scheduler refers to GPS as it processes packets in a fluid manner and not one at a time)

- **PGPS -> Packetized GPS (Weighted Fair Queuing):**

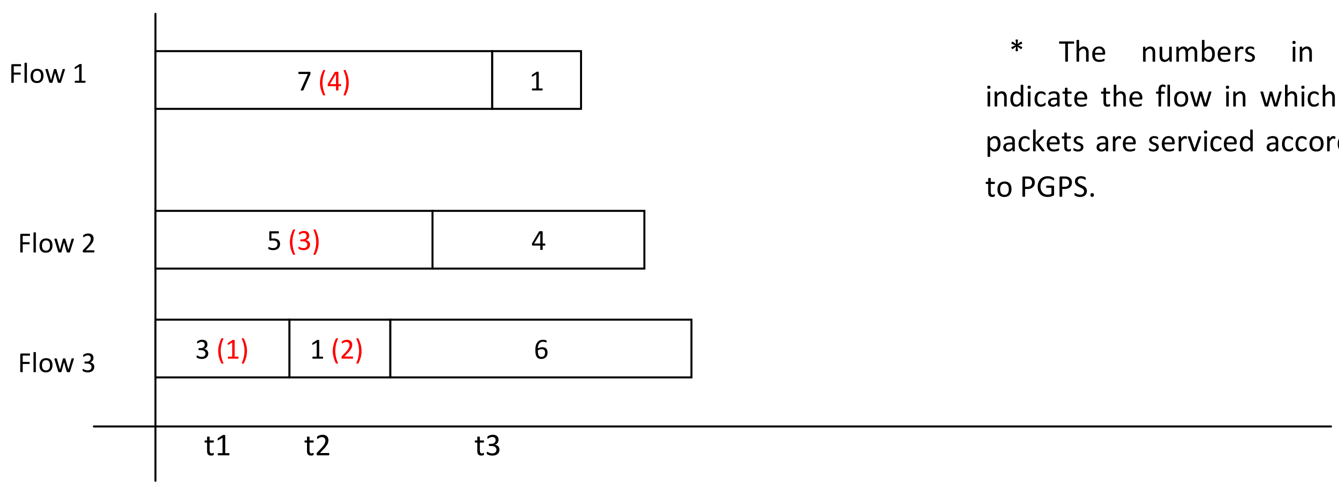
Instead of serving 1 bit at a time, it serves one packet at a time. PGPS is also work conserving scheduler because it won't be idle if there is work to do. Every time the link finishes serving a packet, the scheduler checks if the queue is empty and selects the next packet to serve if the queue is not empty this conserving scheduler.

How do we select the next packet?

> **among all HOL packets,**
> **select the one with the earliest GPS Finish Time**

where, GPS Finish Time is the time in which the packet finishes the idealized GPS scheduler.

Another example:



* The numbers in red indicate the flow in which the packets are serviced according to PGPS.

Assume that all HOL packets arrive at time 0. Further packets arrive at e << 1.
Service rate = 1 bit/sec.

Packets 3,5 and 7 are HOL packets at time 0 and 3 has the least GPS finish time (9 seconds), hence select 3 and now 5,7 and 1 (flow3) will be HOL packets.

Similarly proceed with the other packets. This is a very simple example and more complicated examples will be covered in further lectures.