УДК 621-391

APPS: ADAPTIVE PRIORITY BASED PACKET SCHEDULING METHODE IN IP

SEYED ENAYATALLAH ALAVI[1], MARJAN NADERAN-TAHAN[2], MOHAMAD AMINIAN[3]

[1]*Shahid Chamran University of Ahvaz, Ahvaz, Iran; se_alavi@yahoo.co.uk*
[2]*Shahid Chamran University of Ahvaz, Ahvaz, Iran; m.naderan@scu.ac.ir*
[3]*Shahid Chamran University of Ahvaz, Ahvaz, Iran; m.aminiyan@gmail.com*

We propose adaptive scheduling paradigm that make it possible for IP routers to satisfy different requirements of packet flows. This method promotes conditionally small data packets, which require minor service times, with respect of the maximum delay they can handle that preventing expiration for packets, as long as their total service times cause insignificant delays to other packets in the queue.

Keywords: Artificial intelligence, IP Method, Packet scheduling.

Introduction—Traditional scheduling paradigms of IP networks do not match well the requirements of all categories of packets (for instance sensor packets). Applications which are belong to sensor networks, VoIP and etc. do not really cause significant delays. So naturally prioritized them practically will enhance the overall performance the network and satisfy the needs of this type of packets.

Our primary assumption is that applications like sensor applications generate packets in form of non-congestive traffic. This observation calls for a new metric for application fairness as well, which relies mainly on the delay rather than throughput. In order to avoid the cost of packet preparation for differentiated services, we take advantage of distinctive property of typical non-congestive data:

- *The small size of non-congestive packets*

The key idea of this approach is the service discipline called: "Less Impact Better Service" (LIBS).

II Related works

A lot has been done in the networking community aiming at controlling traffic based on its characteristics. Floyd and Fall introduced mechanisms based on the identification of high-bandwidth flows from the drop-history of RED. The RED-PD algorithm (RED with Preferential Dropping) uses per-flow preferential dropping mechanisms. Two other approaches that use per-flow preferential dropping with FIFO scheduling are Core-Stateless Fair Queuing (CSFQ) and Flow Random Early Detection (FRED). CSFQ marks packets with an estimate of their current sending rate. The router uses this information in conjunction with the flow's fair share estimation in order to decide whether a packets needs to be dropped. FRED does maintain a state although only for the flows which have packets in the queue.

III. Adaptive priority based packet scheduling

First, we assume two different classes of packets(congestive and non-congestive). We use three priority queues. The queue with lower number has higher priority. When packet received by router first based on delay flag we decide to put it in first two high prior queues or the second two queues.

Second step is making decision based on size of packet. So we assume PL (packet length) as critical parameter in our work. Based on what we said the packet with smaller size and lowest flow should receive better services, so we considering two threshold PL1=130 bytes and PL2 as average of length of receiving packets. And we define function PPL as first parameter to decision (if PL2 > PL1 we use (1) but if PL2<PL1 we use (2)):

$$P_{PL} = \begin{cases} 1 & PL \le PL1 \\ \dfrac{(PL - PL1)}{(PL2 - PL1)} & PL1 < PL < PL2 \\ 0 & PL \ge PL2 \end{cases} \qquad (1)$$

Where PL is the packet length if any received pocket.

$$P_{PL} = \begin{cases} 1 & PL \le PL1 \\ 0 & PL \ge PL1 \end{cases} \qquad (2)$$

The second parameter is the priority probability. This probability will be calculated separately for packet that classified in first step and it is called HPP (high priority probability) and LPP (low priority probability):

$$HPP = \frac{number\ pockets\ in\ Queue\ 0}{number\ of\ all\ received\ delay\ sensitive\ pockets}$$
$$LPP = \frac{number\ of\ all\ received\ delay\ sensitive\ pockets}{number\ of\ all\ received\ delay\ sensitive\ pockets} \qquad (3)$$

To continue the first step classification, we integrate two mentioned parameters PPL and HPP/LPP and introduce Final Probability (FP) as an average of the two other probabilities. We use average with same weight. So we define FP as follow :

$$FP = \begin{cases} \dfrac{(P_{PL} + HPP)}{2} & for\ delay\ sensitive\ pockets \\ \dfrac{(P_{PL} + LPP)}{2} & for\ non-delay\ sensitive\ pockets \end{cases} \qquad (4)$$

At the end based on FP, we finalize classification of the received packet. After classification of packets, we need a scheduling method. We choose simple priority scheduling. This method service queues based of their priority.

IV Simulations and results

To test our proposed method we use OPNET 14 simulator. We use dumbbell network topology. We consider the number of delay sensitive flows to the 10 percent of the total flows, and we increased the number of flows in the simulation to examine the network's behavior. The simulations illustrate that our proposed method increases goodput and decrease the overall delay (Fig.1).

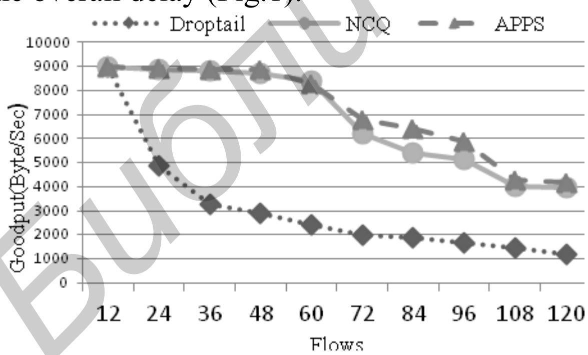

Figure1.a Goodput diagram

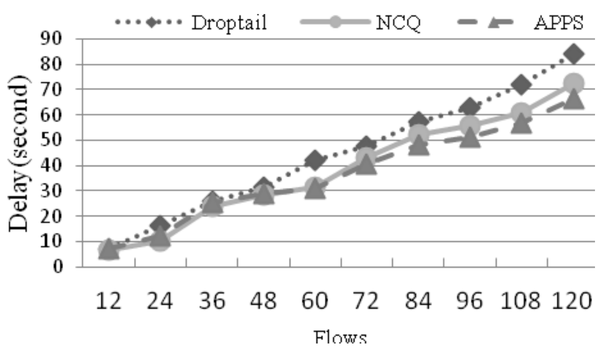

Figure1.b Average overall delay

V. Conclusion

In this work we demonstrate that APPS can be adjusted to promote service for low delay applications. For future works we suggest to do simulations on actual data packets of the applications. Also we suggest performing optimization on parameters using evolutionary algorithms to improve the performance of algorithms.