

Implementing and Testing Priority Scheduler and Token Bucket Policer in Differentiated Service

Ankur Kulhari
Dept. of CSE/IT
JIIT, Noida
Noida, India

Avinash Pandey
Dept. of CSE/IT
JIIT, Noida
Noida, India

Deepshikha Shukla
Dept. of ECE
SRMU, Lucknow
Lucknow, India

ABSTRACT

Internet applications are growing rapidly. The requirement of QoS by these applications varies from very lenient to strict. Maintaining QoS is one of the most typical and challenging task in such scenario. Differentiated services architecture is very popular in such scenario. Differentiated Services is a practical method to implement traffic based service differentiation works on traffic aggregation, per hop behaviour forwarding. Differentiated services uses classifiers to categorize traffic in to flows, policies are defined to allocate the resources to flows and policers are used to shape the bursty traffic whereas schedulers are used to forward the traffic from various traffic queues.

Keywords

DiffServ, IntServ, Policer, Token bucket.

1. INTRODUCTION

When internet was introduced it was never thought that it will become so popular and the density of users will increase so rapid, hence the protocols developed were only meant to address small scale scenario. Provide QoS in today's scenario to various applications over internet using TCP/IP is not sufficient. Traffic over internet is of all categories- some require hard QoS requirements and at the same time other may require no QoS at all. This lead to traffic differentiation in order to allocate network resources according to the QoS required to be maintained.

In early 1990's Integrated services (IntServ) [1] was introduced to support end-to-end quality of service. IntServ works on the concept of RSVP. Resources are pre-reserved while transferring data between two nodes. Resources have to be reserved prior and for every new session this negotiation of resource reservation takes place. Each packet has to process at each intermediate node in order to allocate corresponding resources.

Differentiated services (DiffServ) [2] was introduced aiming to provide more efficient and less complex system for a large internet. DiffServ pushes all complexity to the edge routers by classifying packets at edge routers of an autonomous system. DiffServ provides end-to-end QoS, which is based on PHB aggregation of traffic. Traffic is categorized into flows and these flows were assigned a code point according to which core routers of an autonomous system allocates resources to these flows. In DiffServ complexity is reduced by processing DiffServ Architecture flows instead of handling each packet. All packets of a flow are treated in same way.

Agreements/service provided within a domain

- Service Level Agreement (SLA) with ISP

Edge routers

- Responsible for traffic conditioning
- Perform per aggregate shaping and policing

- Mark packets with a small number of bits; each bit encoding represents a class or subclass

Core routers

- Process packets based on packet marking and defined per hop behavior.

More scalable than IntServ

- No per flow state or signaling

DiffServ architecture–

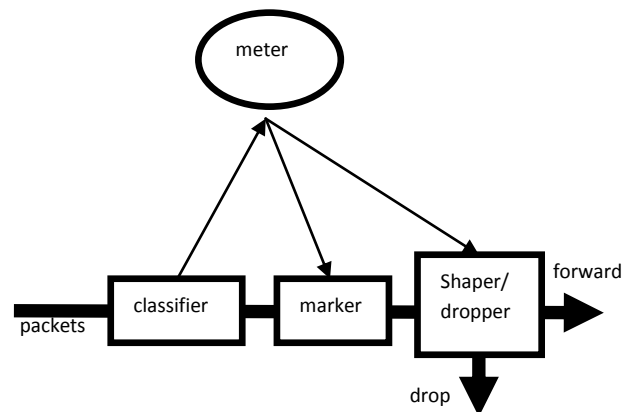


Fig 1: DiffServ Model

Classifier

Classifier is the first module interacts with incoming packets. Classification rules are specified according to various parameters, here src-dest pair is used as classification rules. Classifier marks packets according to classification rules to be specified.

Marker

Profiles are created by classifier, marker marks traffic that falls within profile. Marking is carried out with the help of DS Codepoint, header field of IP packet. The edge router is defined to execute the TSW algorithm as proposed in [3] for marking the packets.

Meter

Metering process helps marker and traffic shaper/dropper to take decisions. It keeps on updating parameters like current rate, current buffer size etc. It checks whether the traffic falls within the negotiated profile.

Scheduler

Once the packets are classified and marked are en-queued into corresponding queues. Schedulers act on en-queued packets to schedule for forwarding.

2. DIFFSERV IMPLEMENTATION WITH PRI SCHEDULING

The simulation tool NS-2 [4,5] is used to simulate the scenario. NS-2 is an open source network simulator freely available on internet. In the carried out simulation PRI is used as the scheduling policy and token bucket is used for classifying the packets and policing the traffic. A modified version of RED [6,7] is used in DiffServ architecture. Traffic is aggregated into different flows i.e. Expedited Forwarding (EF) [8], Assured Forwarding (AF) [9], Best Effort (BE) [5] according to the src-dst mapping and en-queue into corresponding virtual queues.

2.1 Parameters set in simulation

Two physical queues are used with two virtual queues within each physical queue queue (pq_0) is set for EF (Expected Forwarding) flow within which virtual queue (vq_00) is set as the highest priority queue.

Downgraded packets are marked with code point and are en-queued into less priority queues than that of initial codepoint packets, this is done with the help of policer. Token bucket is used as the policer.

2.2 Implementation

```
#Creating links between nodes
$ns duplex-link $src2 $edg1 200Mb 0.1ms DropTail
$ns simplex-link $edg1 $c 50Mb 0.1ms dsRED/edge
$ns simplex-link $cre $edg2 50Mb 0.1ms dsRED/core
#Setting priority queue as scheduler
$qEdg1Cre setSchedulerMode P
#Setting priority of physical queue 0
$qEdg1Cre addQueueRate 0 30000000
```

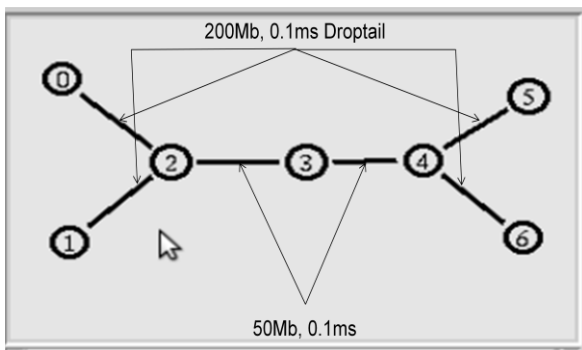


Fig 2: Simulation topology.

```
#Setting number of physical queues
$qEdg1Cre set numQueues_ 2
#Setting number of virtual queues
$qEdg1Cre setNumPrec 2
#Setting parameters for policy table
$qEdg1Cre addPolicyEntry [$src1 id] [$des1 id] TokenBucket
10 $cir1 $cbs1
#Setting parameters for policer table
$qEdg1Cre addPolicerEntry TokenBucket 10 11
#Setting parameters for PHB table
$qEdg1Cre addPHBEntry 10 0 0
#Setting parameters for virtual queues
$qEdg1Cre configQ 0 0 160 180 0.02
Printing current buffer status of virtual queue V_00, in the
file CE_00
puts $CE2_00 "$now [$qCE2 getCurrentV 0 0 ] "
```

2.3 Output

Policy Table(2):
Flow (0 to 5): Token Bucket policer, initial code point 20,

CIR 2500000 bps, CBS 10000 bytes.

Flow (1 to 6): Token Bucket policer, initial code point 10,
CIR 30000000 bps, CBS 30000000 bytes.

Policer Table:

Token Bucket policer code point 10 is policed to code point 11.
Token Bucket policer code point 20 is policed to code point 21.

Per Hop Behaviour Table:

| Code Point | Queue | Precedence |
|------------|-------|------------|
| 10 | 0 | 0 |
| 11 | 0 | 1 |
| 20 | 1 | 0 |
| 21 | 1 | 1 |

Packets Statistics (between edge router Edg1 (node-2) and core router Cre (node-3) in fig. 2)

| CP | TotPkts | TxPkts | ldrops | edrops |
|-----|---------|--------|--------|--------|
| All | 76250 | 62509 | 108 | 13633 |
| 10 | 37500 | 37500 | 0 | 0 |
| 20 | 3134 | 3131 | 3 | 0 |
| 21 | 35616 | 21878 | 105 | 13633 |

Packets Statistics (between core router Cre (node-3) and edge router Edg2 (node-4) in fig. 2)

| CP | TotPkts | TxPkts | ldrops | edrops |
|-----|---------|--------|--------|--------|
| All | 62489 | 62489 | 0 | 0 |
| 10 | 37499 | 37499 | 0 | 0 |
| 20 | 3129 | 3129 | 0 | 0 |
| 21 | 21861 | 21861 | 0 | 0 |

Here, packet statistics shows that vq_01 was not storing any packet for whole simulation, whereas some packets were dropped due to low buffer size from queue vq_10 and vq_11. No packets were dropped at core router traffic shaping took place at edge router only.

Fig. 3 shows the buffer utilization of the queue vq_00. As expected most of the packets were transferred with highest preference, so no packets were delayed.

Fig.4 shows the buffer utilization of the queue vq_01. This queue stores the packets which were dropped from vq_00, but

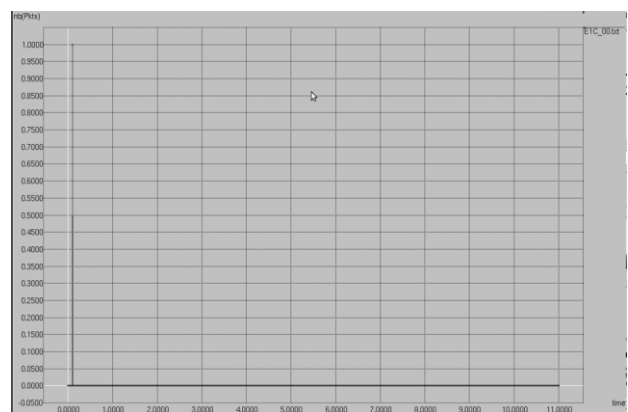


Fig 3: Buffer utilization of the queue vq_00.

here no packets were dropped from vq_00 hence the queue is empty. Vq_01 stores packets which were marked with codepoint 11(downgraded codepoint of 10).

Fig.5 shows the buffer utilization of the queue vq_10. This queue stores the packets of BE(best effort) flow from source nodes1.

Fig. 6 shows the buffer utilization of the queue vq_11. This

queue stores the packets which were dropped from vq_10, vq_10 is not able to accommodate all the packets of BE flow so some packets were assigned downgraded codepoint 21 of codepoint 20 and buffered into vq_11 which is the lowest priority queue. Packets dropped from vq_10 were late dropping no packet marked as early dropped packet.

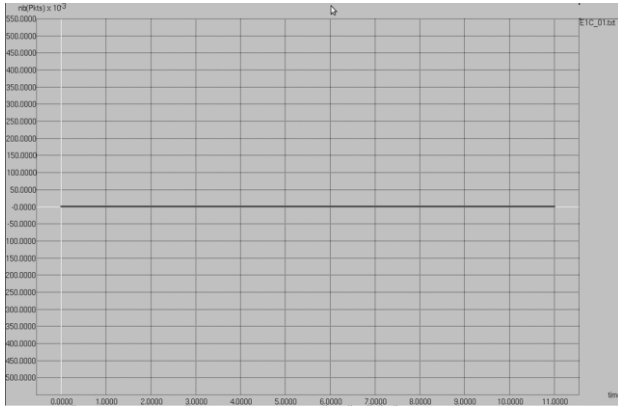


Fig 4: Buffer utilization of the queue vq_01.

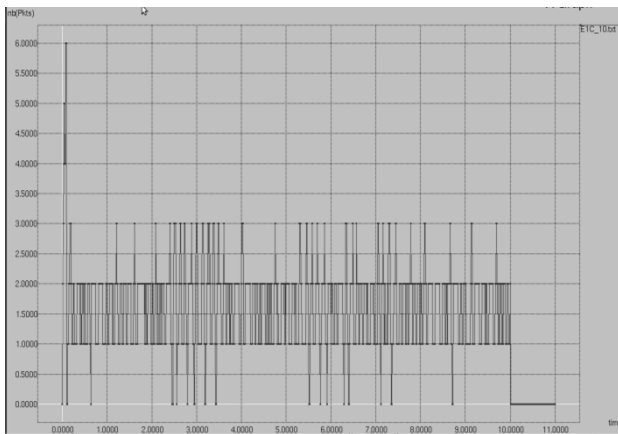


Fig 5: Buffer utilization of the queue vq_10.

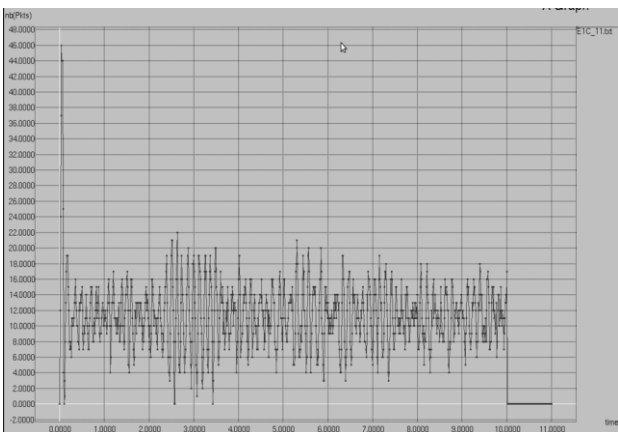


Fig 6: Buffer utilization of the queue vq_11.

3. CONCLUSION

Now days data over Internet is increasing day-by-day in volume as well as categories. We need to provide different QoS parameters for different category of data. From the above work it can be concluded that Differentiated services can be used to provide QoS according to the according to priority of data by categorizing packets into flows. Queue management at DiffServ is an important issue to be handled in order to achieve high throughput.

4. FUTURE WORK

In order to achieve high QoS and efficient buffer management, active queue management plays an important role. Our further research will be mainly focussed on congestion control at Differentiated Services.

5. REFERENCES

- [1] D. Clark, R. Braden, and S. Shenker. Integrated services in the internet architecture: an overview, 1994.
- [2] M. Carlson, W. Weiss, S. Blake, Z. Wang, D. Black, and E. Davies. An architecture for differentiated services. Request for Comments (RFC), 2475, 1998.
- [3] D. D. Clark and W. Fang. Explicit allocation of best effort packet delivery service, *Networking, IEEE/ACM Transactions on*, 6(4): 362–373, 1998.
- [4] Network simulator. <http://www.isi.edu/nsnam/ns/ns-documentation.html>.
- [5] Ns-2 (network simulator). <http://www.isi.edu/nsnam/ns/>.
- [6] B. Pang and W. Gao. Design and performance evaluation of differentiated services router for next generation internet. In *icnmc*, page 316. Published by the IEEE Computer Society, 2001.
- [7] M. A. Qadeer, V. Sharma, A. Agarwal, and S. S. Husain. Differentiated services with multiple random early detection algorithm using ns2 simulator. 2009.
- [8] Poduri K Jacobson V, Nichols K. An expedited forwarding phb. IETF RFC, 2598, June 1999.
- [9] Weiss W Wroclawski J Heinanen J, Baker F. Assured forwarding phb group. IETF RFC, 2597, June 1999.