Differentiated Services

SIGCOMM'99.

Introduction

- There is a growing demand for replacing the current *same-service-to-all* paradigm with a model in which packets, applications, and users are treated differently based on their service needs.
- One way is to replace the current *best-effort* service model with a *reservation-based* service model (or the IntServ Model).
- The problems of the IntServ Model are
  - deployment (not all routers use RSVP or RSVP-enable).
  - scalability (how can we manage so many flows).
- A new approach is the *differentiated service* (DS) model wherein the packet forward behavior (PHB) can be configured

on a per class basis and thereby providing differentiation for large aggregate of network traffics.

- One approach within the DS architecture is to provide the Integrated Services kind of performance measures but without using per-flow state in the network core. This is referred as the *absolute differentiated services*. Examples of this approach are the *premium service* proposed by V. Jacobson and the *assured service* proposed by D. Clark.
- Under the premium service, user can have a performance level that is similar to a leased line as long as the user's traffic is limited to a nominal bandwidth.
- Under the assured service, packets are classified into two levels of drop-preferences (the "in" and "out"classes) at the network edges, depending whether the user follows the allocated bandwidth profile or not.

- A different approach in the DS framework is the *relative differentiated services*. Assume that there are N classes of service, the goal is to guarantee that class i is better (*or at least no worse*) than class $i-1$ for $1 < i \leq N$, in terms of local (per hop) performance metrics for the queueing delays and packet loss.
- Under the relative differentiated model, applications and users do not get an absolute service level assurance, such as the end-to-end delay bound or bandwidth.

- There are several ways to implement the relative differentiation model, the are:
  - *Strict Prioritization:* although this method can provide quality ordering, it has some major drawbacks like: (1) If the higher classes are persistently backlogged, it can result in long starvation periods for lower classes. (2) It is not *controllable*, that is, it does not provide tuning knobs for adjusting the quality spacing.
  - *Price Differentiation:* Assume that higher prices will lead to lower loads in the higher classes and thus better service quality. The major drawback is that this can only be effective over relatively long timescales. Also, when higher classes get overloaded (e.g., many rich users become active), they will offer worse packet forwarding than lower classes.
  - *Capacity Differentiation:* Allocate larger amount of forwarding

resources (e.g., bandwidth or buffer space) to higher classes, relative to the expected load in each class. One good example of this is the weighted fair queueing (WFQ) with class i having a weight of w_i, that is:

$$\frac{w_i}{\lambda_i} > \frac{w_j}{\lambda_j} \qquad \text{if } i > j$$

where λ_x is the average arrival rate of class x. Major drawback is that in *shorter* time-scales, the class ordering is *not* maintained. This is very important because many Internet applications are of short duration (e.g., a Web session).

- In the SIGCOMM'99 paper, the authors only considered "delay differentiation".

The Proportional DiffServ Model

- The goal is to "space" certain class performance metrics proportionally to the differentiation parameters that the network operator chooses. Let q_i is the performance measures for class i and c_i is the quality differentiation parameters for class i. We have:

$$\frac{q_i}{q_j} = \frac{c_i}{c_j} \qquad i, j = 1, 2, ..N$$

  where $c_1 < c_2 < \cdots < c_N$.

- Important points are:
  - the *ratio* should be kept irregardless of the loading variation of the individual classes.

  - the *ratio* should be kept irregardless of the time scales (e.g, a few packet transmission times, or hundred of packet transmission times).

- A formulation of the proportional differentiation model for a short-term queueing metrics. Let $\bar{d}_i(t, t+\tau)$ be the average queueing delay of the class i packets departing in the interval $(t, t+\tau)$ where $\tau > 0$ is the *monitoring timescale*. If there are no departing packets, $\bar{d}_i(t, t+\tau)$ is undefined (????) in this interval. The service model is:

$$\frac{\bar{d}_i(t, t+\tau)}{\bar{d}_j(t, t+\tau)} = \frac{\delta_i}{\delta_j}$$

  for all time intervals $(t, t+\tau)$ in which both $\bar{d}_i(t, t+\tau)$ and $\bar{d}_j(t, t+\tau)$ are defined and $\delta_1 > \delta_2 > \cdots > \delta_N > 0$.
  (Note: the author claimed that they cannot derive the feasibility conditions for this short timescales formulation.)

Dynamics and Feasibility of Prop. Delay Diff.

- Let us study and understand the dynamics and under what conditions in which this model is feasible, for the case of long-term average delay.
- Only consider *lossless* and *work-conserving* packet scheduler.
- Let λ_i be the average arrival rate in class i and $\lambda = \sum_{i=1}^{N} \lambda_i$ be the aggregated average arrival rate. Assume there exists a scheduling discipline which can enforce the proportional delay differentiation (PDD) for long-term average queueing delay, that is:

$$\frac{\bar{d}_i}{\bar{d}_j} = \frac{\delta_i}{\delta_j} \quad i, j = 1, 2, ...N \tag{1}$$

- If the packet length distribution is the same in all classes, the conservation law states that:

$$\sum_{i=1}^{N} \lambda_i \bar{d}_i \quad = \quad \lambda \bar{d}(\lambda) \tag{2}$$

  where $\bar{d}(\lambda)$ is the average queueing delay that would result if the aggregated traffic was serviced by a work-conserving FCFS server of the same capacity. Note that $\bar{d}(\lambda)$ is a function of the aggregated arrival rate.

- Combine the previous two equations, we have:

$$\bar{d}_i \quad = \quad \frac{\delta_i \bar{d}(\lambda)}{\sum_{k=1}^{N} \delta_k \lambda_k / \lambda} \tag{3}$$

- We have the following properties:
  - $\bar{d}_i$ increases with the arrival rate of each class j.
  - Increasing the load of a higher class causes a larger increase in the average delay of a class than increasing the load of a lower class.
  - If the delay differentiation parameter of a class increases, the average delay of all other classes decreases while the average delay of that class increases.
  - Suppose that a fraction of the class i load switches to class j while the aggregate load remains the same, the average delay of each class increases if $i < j$, and decreases if $i > j$.
- Comment on the **feasibility**.

Two Packet Schedulers for Relative Delay Diff.

The two schedulers approximate the PDD model in heavy-load conditions and later simulation show that even in short timescales.

The two schedulers are:

- Backlog-Proportional Rate (BPR) Scheduler
- Waiting Time Priority (WTP) Scheduler

Backlog-Proportional Rate (BPR) Scheduler

- Use the bandwidth distribution model of a GPS server.
- The main idea is to dynamically readjust the class service rate so that they are always ratioed proportionally to the corresponding ratios of class loads.
- What is a class load? It is reflected on the relation of the class backlogs.
- Specifically, let $r_i(t)$ be the service rate that is assigned to class i at time t. If class i has an empty queue, then $r_i(t) = 0$. For two backlogged classes i and j, the service rate allocation in BPR satisfies the proportionality constraint:

$$\frac{r_i(t)}{r_j(t)} = \frac{s_i}{s_j}\frac{q_i(t)}{q_j(t)} \quad (4)$$

where $q_i(t)$ is the backlog of class i at time t. The actual service rate of each class during a busy period can be calculated from the work-conserving constraint;

$$\sum_{i=1}^{N} r_i(t) = R$$

where R is the link capacity.
The parameters $\{s_i\}$ are the *scheduler differentiation parameters* (SDPs) that the network operator selects wherein $s_1 < s_2 < \cdots < s_N$.

- The simulation shows that the DDP ratios tending to the inverse of the corresponding SDP ratios in heavy load conditions, that is:

$$\frac{\bar{d}_i}{\bar{d}_j} \quad \rightarrow \quad \frac{\delta_i}{\delta_j} = \frac{s_j}{s_i} \tag{5}$$

Note: No mathematical or analytical justification.

- One observation of the BPR scheduler is that it will experience a sawtooth-type of delay variation. This is true because of a queue with only a few packets left, it will have smaller service rates and therefore these packets will experience a higher delay.
- Note that BPR scheduler is only a "theoretical" scheduler (because it is based on the fluid server model). In practice, one needs to "approximate" it using a packetized version of BPR.

Waiting-Time Priority (WTP) Scheduler

- The main idea is that the priority of a packet ($p_i(t)$) increases proportionally with its waiting-time. That is:

$$p_i(t) \quad = \quad w_i(t)s_i \tag{6}$$

  where $w_i(t)$ is the waiting-time of the packet at time t.

- The main finding is that WTP approximates the proportional delay differentiation model in heavy load conditions, with the DDP ratios tending to the inverse of the corresponding SDP ratios.

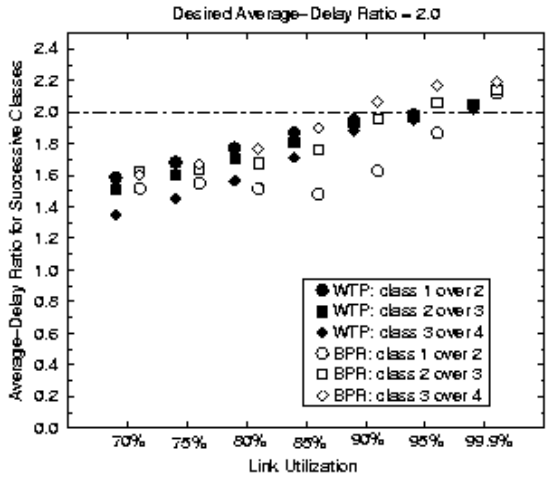
- The simulation also show that WTP scheduler achieves this goal more accurately than the BPR scheduler.

- the WTP also approximate the short timescale of the proportional differentiation model in time intervals of high load, even when the monitoring timescale τ is a few tens of packet transmission times.
- WTP also exhibits a problematic behavior in short timescales. Under certain conditions on the SDPs, an arriving burst in a high-class queue can exclude lower classes from service until this burst is completely serviced.
- AGAIN, no mathematical and analytical basis.

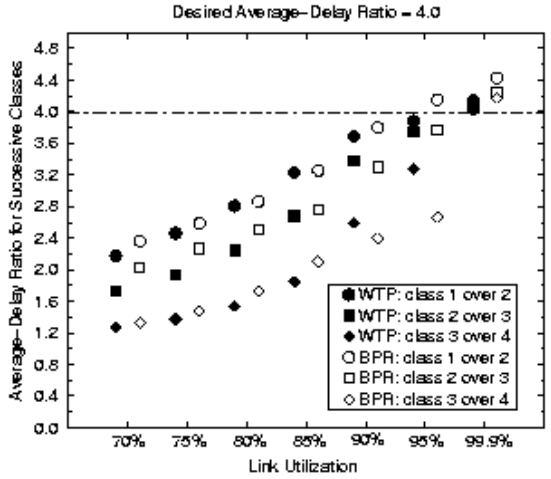
Simulation

- $N = 4$ classes. $s_1 = 1, s_2 = 2, s_3 = 4, s_4 = 8$.
- Interarrivals between packets of the same class follow a Pareto distribution with shape parameter $\alpha = 1.9$.
- Class load distribution is $[40, 30, 20, 10]$.
- Packet length distribution is the same for all classes wherein 40% are 40 bytes, 50% are 550 bytes and 10% are 1500 bytes.
- Average packet transmission time is 11.2 units.
- Examine at moderate load (ρ is around 0.7) and high load (ρ is around 0.95).

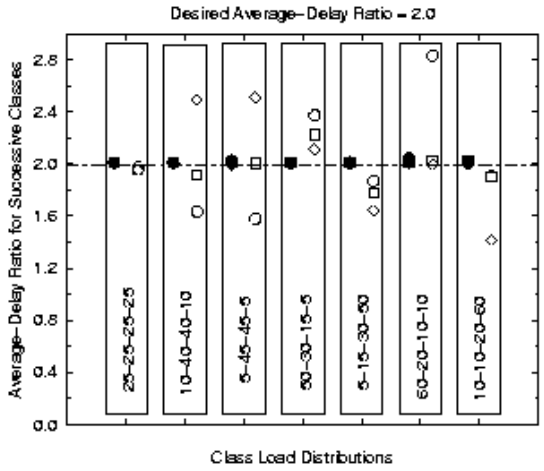
- Figure 1 and 2, large timescales. (1) As the aggregate load increases, the WTP scheduler tends to proportional average-delay differentiation. (2) BPR has similar trend but it does not converge exactly to the specified ratio (may be due to packetization).
- Figure 3. Performance of algorithms at different time scales.
- Figure 4 and 5, microscopic view of BPR and WTP.
- Extension to multi-hop multi-classes traffic.



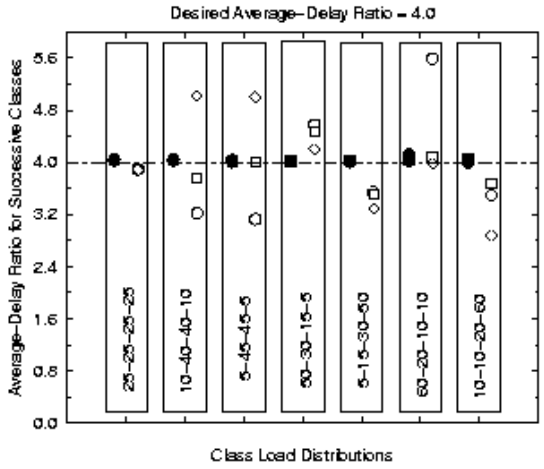
(a) $s_1 = 1, s_2 = 2, s_3 = 4, s_4 = 8$



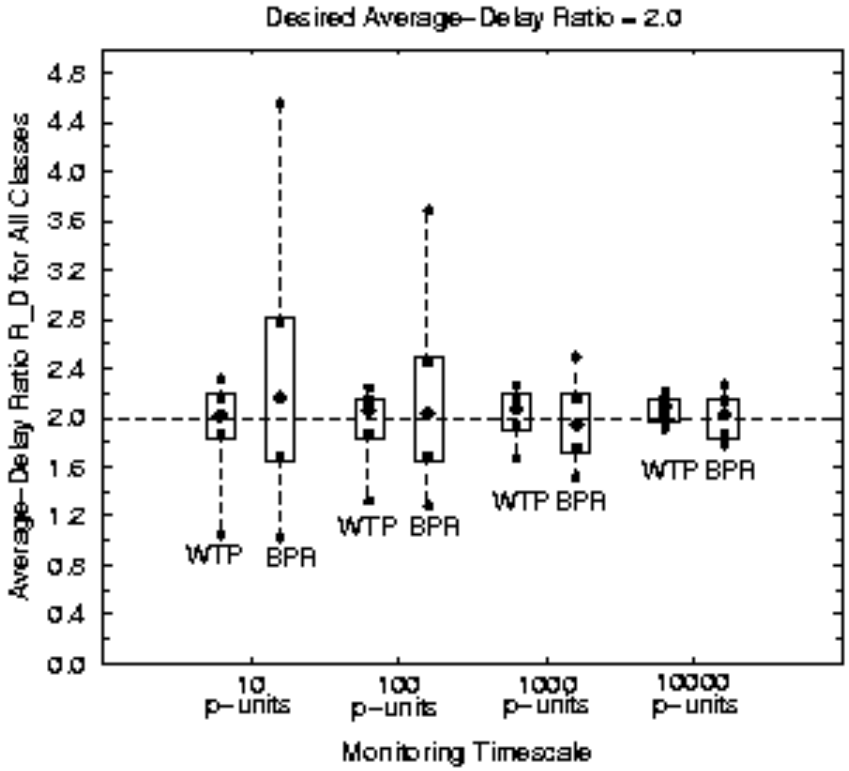
(b) $s_1 = 1, s_2 = 4, s_3 = 16, s_4 = 64$



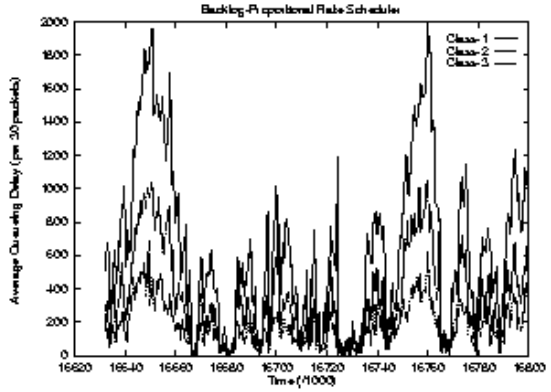
(a) $\delta_1 = 1, \delta_2 = 2, \delta_3 = 4, \delta_4 = 8$



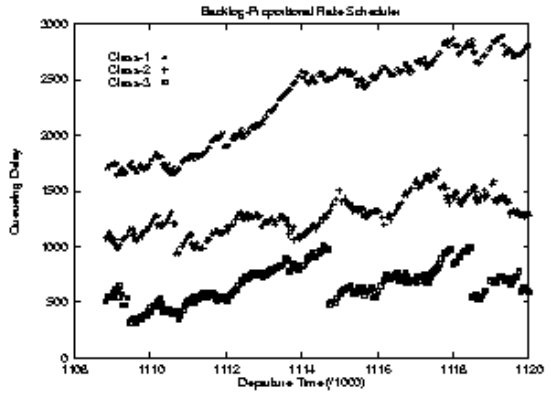
(b) $\delta_1 = 1, \delta_2 = 4, \delta_3 = 16, \delta_4 = 64$



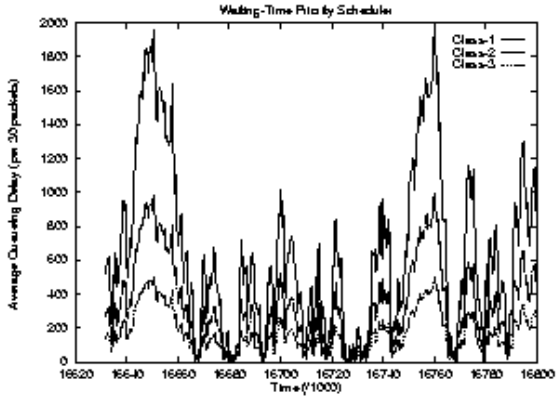
(a) $s_1 = 1, s_2 = 2, s_3 = 4, s_4 = 8$



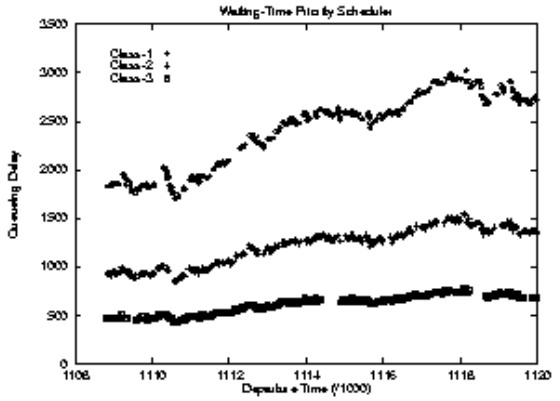
(a) Microscopic view I



(b) Microscopic view II



(a) Microscopic view I



(b) Microscopic view II