Differentiated Service Routers in the Internet

Differentiated Service (DiffServ)

- Quality of Service Architectures in the Internet:
  - Integrated Services (IntServ)
  - Differentiated Services (DiffServ)
- DiffServ: Differrent packet treatment depending on packet marking in the IP-Header (DS-Field)
- Move complexity to the edge routers and keep processing in the core routers as simple as possible
- Packet marking and remarking performed at:
  edge routers (between subscribers),
  border routers (between provider domains)
  and applications

Proportional Differentiated Services:

- **Two approaches to service differentiation:**
  - Absolute Differentiated Services: provides guarantees and statistical assurances to performance measures, e.g.: E2E delay
  - Relative Differentiated Services: ordering of a number of traffic classes in terms of their packet forwarding behaviour
- **Proportional Differentiated Services:**
  - further refinement of the relative Differentiated Services approach
  - quantifies the spacing between the traffic classes
- **Possible differentiation criteria:**
  - throughput
  - delay
  - jitter
  - loss rate

Proportional Delay Differentiation:

- Definition of N traffic classes
- Differentiation criteria: queueing delay
- Definition of the ratios of mean delay of consecutive classes

$$\frac{E[W_i]}{E[W_j]} = \frac{\delta_i}{\delta_j}.$$

- Class differentiation parameters (delay differentiation parameters: DDP)

$$\delta_1 > \delta_2 > .... > \delta_N$$

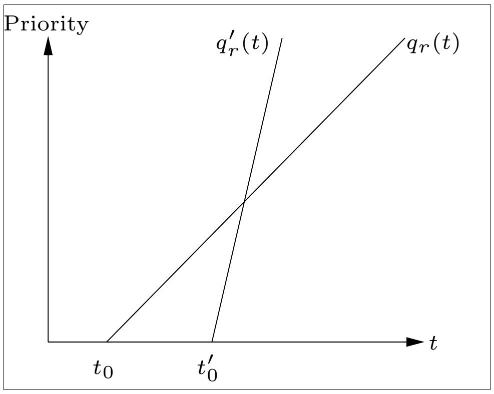
so that higher classes are "better" than lower classes

Dynamic Scheduling Model

- Priority Function

$$q_r(t) = (t - t_0) \cdot b_r$$

$$0 \leq b_1 \leq b_2 \leq \ldots \leq b_r$$



➤ Priority of a class-r packet increases linearly with the elapsed time in the queueing system at a rate $\boldsymbol{b_r}$

◆ Main Waiting Time:

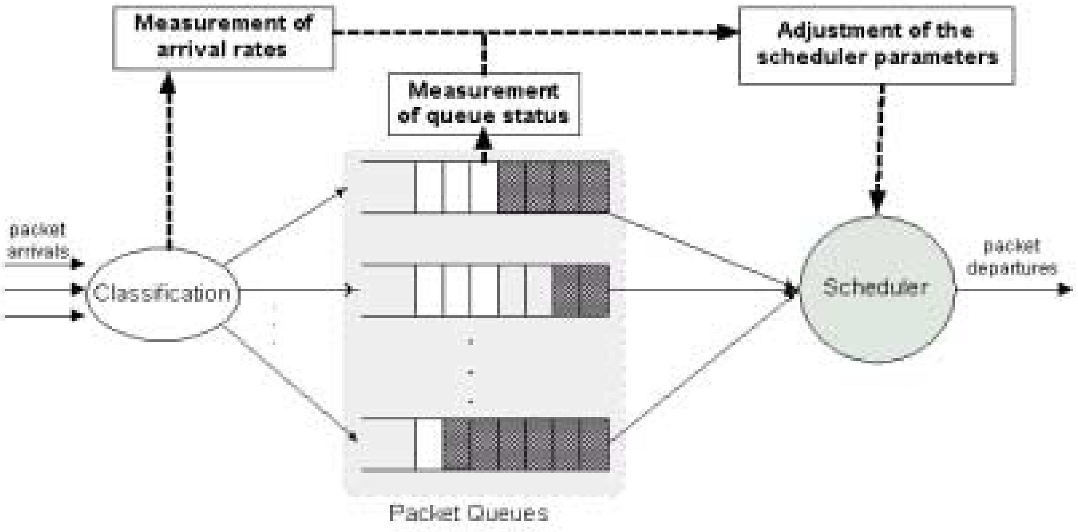
$$\overline{W}_r = \frac{\dfrac{\overline{W}_0}{1-\rho} - \sum_{i=1}^{r-1} \rho_i \overline{W}_i \left(1 - \dfrac{b_i}{b_r}\right)}{1 - \sum_{i=r+1}^{R} \rho_i \left(1 - \dfrac{b_r}{b_i}\right)}$$

◆ Extended Priority Function:

- ➤ dependency on the current load ρ_i
- ➤ dependency on the delay differentiation parameters (DDPs) δ_i

$$q_r(t) = (t - t_0) b_r(\rho_1, \ldots, \rho_R, \delta_1, \ldots, \delta_R), \qquad 1 \le r \le R$$

Dynamic Priority Scheduling - Architecture



Optimization Problem:

- R traffic classes
- Given DDPs: $\delta_1 > \delta_2 > ... > \delta_R$
- Measured class utilizations: $\rho_1 > \rho_2 > ... > \rho_R$
- Ideally:

$$\sum_{i=1}^{R-1} \left(\frac{E[W_i]}{E[W_{i+1}]} - \frac{\delta_i}{\delta_{i+1}} \right)^2 = 0,$$

- Or:

$$\sum_{i=1}^{R-1} g_i \cdot \left(\frac{E[W_i]}{E[W_{i+1}]} - \frac{\delta_i}{\delta_{i+1}} \right)^2 \rightarrow min$$

Example of two Traffic Classes:

- Exact form to calculate the slopes of the priority functions:

$$b_1 = 1$$

$$\frac{b_1}{b_2} = 1 - \frac{1}{\rho}(1 - \frac{\delta_2}{\delta_1})$$

- Feasability:

$$\rho > 1 - \frac{\delta_2}{\delta_1}$$

- Limit: As the load aproaches 100%, the ratio of the scheduler parameters: b_1/b_2 tends to the inverse of the DDPs: δ_2/δ_1

Optimization using Genetic Algorithms

- Genetic algorithms(GAs) are optimization and search procedures inspired by genetics and the process of natural selection
- Structure of a genetic algorithm
  - Randomly generate a population of individuals (bit strings).
  - Decode each individual and evaluate its fitness.
  - Generate a new population partly by cloning, partly by combining and partly by mutating the fittest individuals.
  - Repeat steps 2 and 3 until a stop condition holds.

Example: 4 Traffic Classes, Required Delay Ratio: 2

- Scheduler Prameters: b_1, b_2, b_3, b_4

ρ [in %]	b_1	b_2	b_3	b_4
70%	1.00	6.42	13.10	9999792
75%	1.00	6.53	20.61	999807
80%	1.00	3.61	16.06	218.29
85%	1.00	2.83	8.54	34.36
90%	1.00	2.42	6.01	16.72
95%	1.00	2.17	4.75	10.82
100%	1.00	2.00	4.00	8.00

◆ Mean Delays and Delay Ratios

ρ [%]	$E[W_1]$ [msec]	$E[W_2]$ [msec]	$E[W_3]$ [msec]	$E[W_4]$ [msec]	$\frac{E[W_1]}{E[W_2]}$	$\frac{E[W_2]}{E[W_3]}$	$\frac{E[W_3]}{E[W_4]}$
70%	6.465	3.232	2.424	1.212	2.00	1.33	2.00
75%	8.415	3.892	2.462	1.231	2.16	1.58	2.00
80%	10.667	5.333	2.667	1.333	2.00	2.00	2.00
85%	14.222	7.111	3.556	1.778	2.00	2.00	2.00
90%	21.333	10.667	5.333	2.667	2.00	2.00	2.00
95%	42.666	21.334	10.667	5.333	2.00	2.00	2.00
100%	21331	10667	5334	2668	2.00	2.00	2.00

Implementation

◆ Offline calculation of the scheduling parameters

◆ Dynamic adjustment during router runtime operation

◆ Adjustment:

- ➤ Look-up tables: pre-calculated parameter sets for intervals of the utilization range --> discretization error
- ➤ Use of simple interpolation functions

◆ Adjustment trigger:

- ➤ Periodical
- ➤ On special events: significant utilization change in one or more traffic classes

Experiments with other Priority Functions

- **Motivation**
  - Investigate the performance using other priority functions for lower utilizations (especially around 70%)
  - Using initial priorities could better fulfill the proportional differentiation requirements, as the priority initially assigned to the arriving packets is higher than zero
- **Priority Function with Initial Priorities**
  - Higher optimization error (partially due to the approximate formulae)
- **Quadratic Priority Function**
  - No significant improvement in comparison to the linear priority function

Conclusion

Approach of Dynamic Priority Scheduling

- Applies equally well to heavy and moderate load conditions
- Offline determination of the scheduling parameters and table look-up or simple interpolation functions during runtime
- Use of genetic algorithms for optimization

Ongoing work:

- Simulation to validate developed results
- Impact of infinite buffer approximation and exponential distribution
- Further consideration of new Internet traffic modelling methods (Self similar traffic, Pareto distribution)
- Optimization using simulation for nonexp. distribution