Adaptive Scheduling for quality differentiation

Johanna Antila

Networking Laboratory, Helsinki University of Technology

{jmantti3}@netlab.hut.fi

Outline

- Introduction
- Contribution
- Differentiation models
- Packet schedulers
- Simulation model
- Simulation results
- Conclusions
- Current and future work

Introduction

- The Internet has developed from a research network into a multiservice network
  - diverse applications and customers
- New QoS schemes are required
  - Packet scheduler is a key component in QoS provisioning
    - shares the common resources by deciding the order at which packets are served

Contribution

- Starting point:
  - Service differentiation is based on DiffServ architecture
- We study two important differentiation models
  - Capacity and delay differentiation
- We propose schedulers for implementing these models
- By simulations we evaluate
  - The viability of the differentiation models
  - Performance of the proposed schedulers

Differentiation models

- Two differentiation models are examined:
  - Absolute capacity differentiation
  - Proportional delay differentiation with delay bound
- In proportional models the highest class is assigned with a delay bound
  - This is because proportional models as such are not able to guarantee small delays

Differentiation models

- Notations:
  - w_i = weight of class i
  - g_i = guaranteed rate of class i
  - C = link capacity
  - δ_i = differentiation parameter
  - d_i = average queuing delay of class i

Differentiation models

- Absolute capacity differentiation:
  - each service class is allocated a predefined amount of link capacity, determined by the class weight w_i.
  - In an ideal case, class *i* should receive service in any interval (τ, t) with a rate

$$g_i \geq \frac{w_i}{\sum_{j=1}^{N} w_j} C$$

Differentiation models

- Proportional delay differentiation:
  - the ratio of average queuing delays in any two classes *i* and *j* should equal the ratio of differentiation parameters in these classes for the interval (τ, t) :

$$\frac{d_i(\tau, t)}{d_j(\tau, t)} = \frac{\delta_i}{\delta_j}$$

Packet schedulers

- The differentiation models were implemented with the following schedulers

Packet scheduler	Quality parameter	Differentiation model
DRR	Capacity	Absolute
ADRR with delay bound	Delay	Proportional with delay bound
HPD with delay bound	Delay	Proportional with delay bound

Packet schedulers

- Notations:
  - w_i = weight of class i
  - $q_i(t)$ = filtered queue length of class i at time t
  - $d_i(t)$ = average delay of class i at time t
  - $w_i(t)$ = head waiting time of class i at time t
  - δ_i = differentiation parameter of class i
  - g = constant

Packet schedulers

- DRR scheduler:
  - aims at approximating an ideal, fluid based GPS scheduler
- Each class is assigned with a weight w_i
  - In each service round, a frame of N bits is divided among the classes in proportion to the weights
  - Provides fairness also when variable size packets are used

Packet schedulers

- Adaptive DRR scheduler (ADRR)
  - aims to provide proportional delay differentiation. Furthermore, we have assigned the highest class with a delay bound
- The weights for the interval (τ, t) are updated in the following way:

$$w_i(t) = \frac{q_i(t)}{\sum_{k=1}^{n} \frac{\delta_i}{\delta_k} q_k(t)}$$

Packet schedulers

- HPD scheduler
  - also aims to provide proportional delay differentiation. Again, we have assigned the highest class with a delay bound.
  - When the server becomes free, HPD selects for transmission a packet from a backlogged class *j* with maximum normalized hybrid delay:

$$j = \arg\max(g d_i(t) / \delta_i + (1 - g) w_i(t) / \delta_i)$$

Simulation model

- A specific simulator was implemented with CNCL
  - CNCL is a freeware C++ class library package
  - It consists of basic functionality required to support event-driven simulation
  - The user has to implement most of the functionality by herself

Simulation model

- The simulation model consisted of the following components:
  - Node and link models
  - Simple traffic generator models
    - Control traffic
    - VoIP
    - Video (short flows)
    - WWW
    - FTP
  - Simple TCP model (including slow start and RTT estimation)

Simulation model

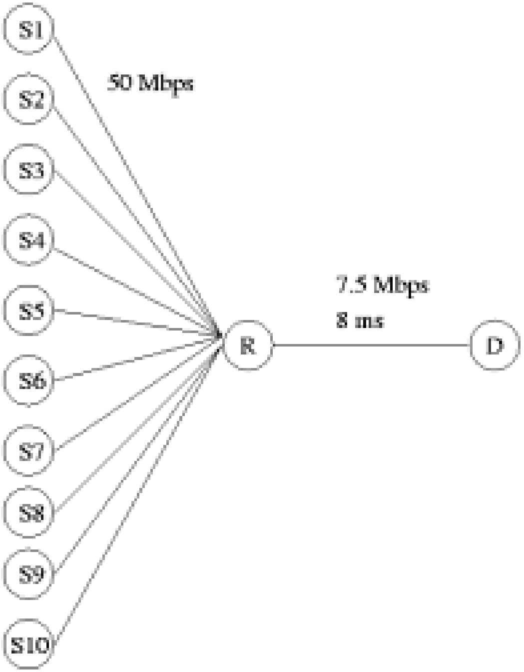
- Baseline:
  - A best effort scenario with FCFS scheduler
- Then, simulations were performed in eight scenarios for each scheduler:
  - Four scenarios where different traffic types were separated based on some criteria (transport protocol, application type etc.)
  - Four scenarios where different traffic types were allowed to be mixed.

Simulation model

- Provisioning rules for the schedulers:
  - DRR:
    - real time traffic was provisioned two times the expected load share and the remaining capacity was divided between other classes in proportion to their load shares
  - HPD and Adaptive DRR with delay bound:
    - Delay bound for the highest class was set to 5 ms, delay ratio between other classes was set to 4.
  - Queue management method was TailDrop

Simulation model

- The following topology was used in the simulations:



Simulation results (DRR)

- In the table below results are shown when traffic is mixed
  - Only minor difference between throughputs and delay of WWW sessions of different classes
  - Huge losses especially for WWW

		Queueing delay		Throughput		Loss	
Traffic	**Class**	**Mean**	**Stdev**	**Mean**	**Stdev**	**Mean**	**Stdev**
FTP	0	196 ms	152 ms	1170370 bps	558410 bps	0.3 %	1.5 %
WWW	1	19 ms	17 ms	173090 bps	387350 bps	0.01 %	0.5 %
WWW	2	22 ms	16 ms	185270 bps	359040 bps	7.4 %	11.9 %
Video	2	19 ms	5 ms	481490 bps	16760 bps	3.7 %	3.4 %
VoIP	3	2 ms	0 ms	30450 bps	6130 bps	0 %	0 %
Control	3	3 ms	0 ms	71250 bps	0 bps	0 %	0 %

Simulation results (DRR)

- In the following table different traffic types are separated
  - losses are smaller
    - however, FTP suffers from overprovisioning for real-time traffic

		Queueing delay		Throughput		Loss	
Traffic	**Class**	**Mean**	**Stdev**	**Mean**	**Stdev**	**Mean**	**Stdev**
FTP	0	335 ms	259 ms	865480 bps	458460 bps	1.2 %	3.8 %
WWW	1	44 ms	37 ms	131220 bps	313100 bps	0.8 %	3.9 %
Video	2	7 ms	7 ms	493190 bps	13140 bps	1.4 %	2.6 %
VoIP	3	2 ms	0 ms	30210 bps	6170 bps	0 %	0 %
Control	3	3 ms	0 ms	71250 bps	0 bps	0 %	0 %

Simulation results (ADRR)

- In the table below results are shown for ADRR when traffic is mixed
  - Better differentiation compared with DRR
    - Delay bound is met but target ratios are not
    - Quite high losses due to weight adaptation

		Queueing delay		Throughput		Loss	
Traffic	Class	Mean	Stdev	Mean	Stdev	Mean	Stdev
FTP	0	284 ms	122 ms	1626380 bps	583450 bps	1.0 %	3.7 %
FTP	1	167 ms	48 ms	1440270 bps	842350 bps	5.2 %	7.2 %
WWW	1	50 ms	55 ms	146700 bps	351600 bps	0.9 %	3.9 %
WWW	2	22 ms	18 ms	192130 bps	378980 bps	8.8 %	13.3 %
Video	2	17 ms	7 ms	479790 bps	16090 bps	4.0 %	3.2 %
VoIP	3	4 ms	1 ms	30160 bps	5550 bps	0 %	0 %
Control	3	4 ms	0 ms	71250 bps	0 bps	0 %	0 %

Simulation results (HPD)

- The table below shows the results for delay bounded HPD when traffic is separated
  - Both delay bound and delay ratios are met
    - FTP does not suffer so much, because overprovisioning for real-time traffic is not required

		Queueing delay		Throughput		Loss	
Traffic	**Class**	**Mean**	**Stdev**	**Mean**	**Stdev**	**Mean**	**Stdev**
FTP	0	300 ms	132 ms	1345400 bps	613160 bps	1.3 %	4.1 %
WWW	1	67 ms	54 ms	119480 bps	303010 bps	1.9 %	6.0 %
Video	2	17 ms	7 ms	498950 bps	2050 bps	0.2 %	0.3 %
VoIP	3	4 ms	1 ms	30220 bps	6340 bps	0 %	0 %
Control	3	5 ms	0 ms	71250 bps	0 bps	0 %	0 %

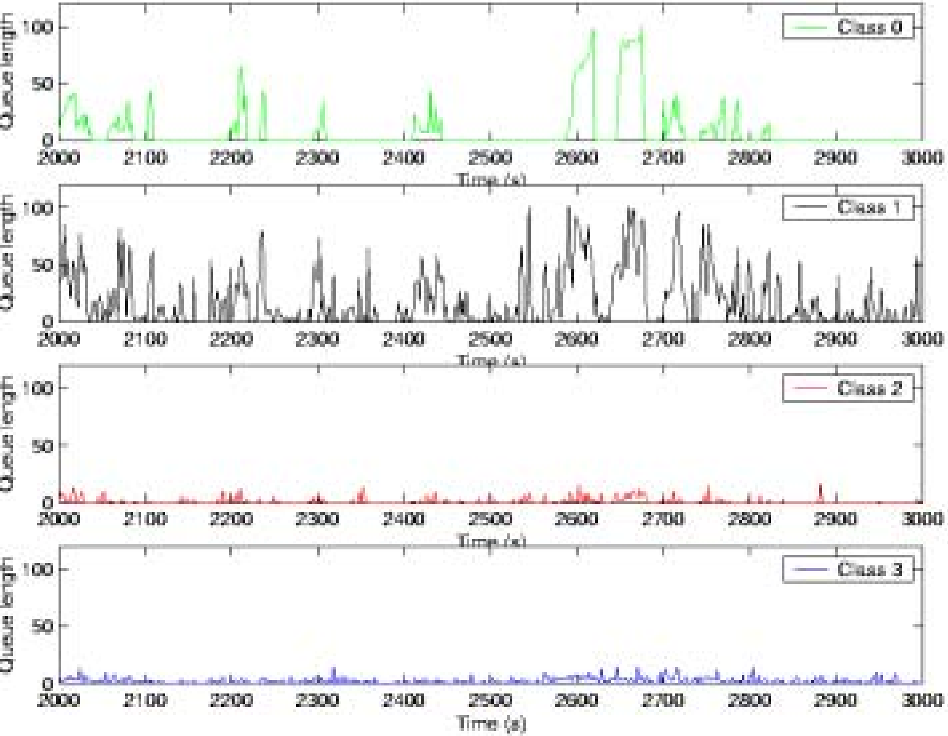
Simulation results (HPD)

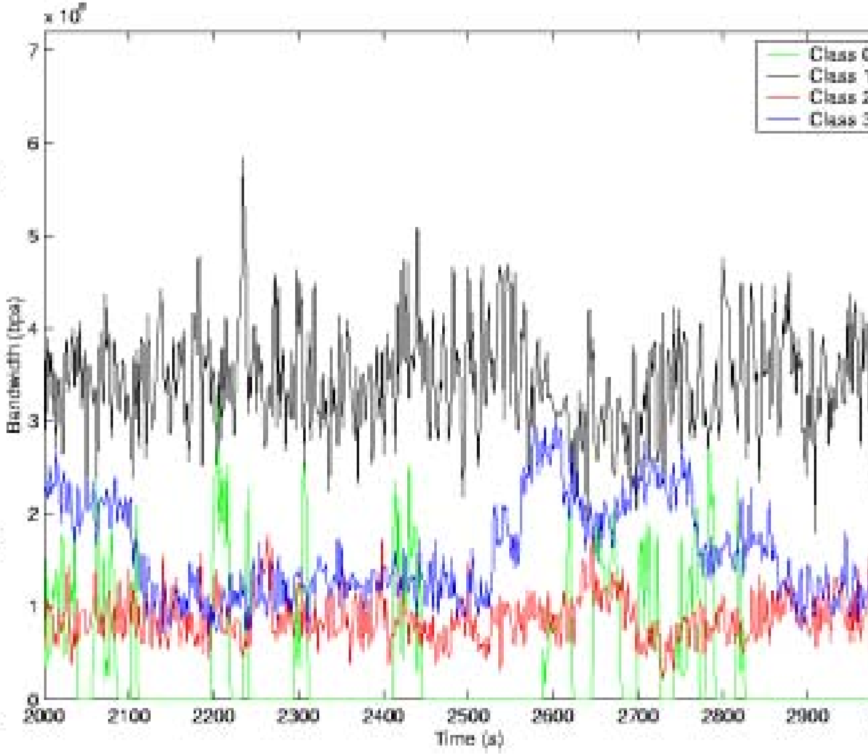
- When different traffic types are mixed
  - Delay bound and ratios are still met
    - However, losses become intolerable

		Queueing delay		Throughput		Loss	
Traffic	Class	Mean	Stdev	Mean	Stdev	Mean	Stdev
FTP	0	276 ms	175 ms	1211400 bps	547050 bps	0.7 %	3.0 %
FTP	1	156 ms	43 ms	1557670 bps	676790 bps	4.0 %	3.9 %
WWW	1	63 ms	54 ms	120880 bps	303790 bps	0.8 %	4.0 %
WWW	2	17 ms	14 ms	182010 bps	393090 bps	2.9 %	7.3 %
WWW	3	6 ms	2 ms	220470 bps	451800 bps	1.0 %	3.2 %
Video	2	17 ms	9 ms	493700 bps	5190 bps	1.3 %	1.0 %
Video	3	5 ms	0 ms	499320 bps	350 bps	1.4 %	0 %
VoIP	3	4 ms	1 ms	30290 bps	6250 bps	0.1 %	0.1 %
Control	3	5 ms	0 ms	71030 bps	0 bps	0.5 %	0 %

Simulation results (HPD)

- Bandwidth allocation follows queue lengths





Conclusions

- From applications point of view it is beneficial to separate different traffic types:
  - two classes for TCP traffic: one for short flows, one for long flows
  - one or two classes for real time traffic: streaming type traffic and VoIP etc.
- Differentiation and provisioning with static schedulers (DRR) is problematic
  - measurement based schedulers are more suitable for changing load conditions

Conclusions

- Schedulers for proportional delay differentiation have to be integrated with a delay bound for the highest class
  - HPD with delay bound was best able to meet the differentiation target due to its robust delay estimator
    - however, if traffic is mixed arbitrarily, losses become intolerable

Current and future work

- A simulation environment in ns2 has been constructed
  - more accurate traffic models (full-tcp, MPEG4 traffic etc.)
- With this simulation environment we aim to
  - verify the results from previous research
  - investigate larger network topologies: end-to-end aspect
  - investigate intra-class performance
  - study the effect of different active queue management and policing mechanisms

Current and future work

- Future work will also include
  - Further development of the algorithms and measurement based estimators
  - Implementation and measurements of the delay-bounded HPD algorithm in the prototype environment