A Case for Relative Differentiated Services and the Proportional Differentiation Model

Constantinos Dovrolis

Parameswaran Ramanathan

University of Wisconsin-Madison

IEEE Network Journal

September-October 1999

Outline

- Need for differentiated services
  - Very diverse quality of service expectations
  - Current internet has same-service-to-all model
  - Classify the users based on their needs
- Differentiated services approaches
  - Absolute differentiated services
  - Relative differentiated services
  - Proportional differentiated services

Outline

- Absolute differentiated services : provides guarantees for absolute performance
- Relative differentiated services : segregation into service classes based on quality and pricing constraints
- Proportional differentiated services : strict quality spacing between adjacent classes

Internet Users and Applications

- Companies, elite users – willing to pay a higher cost for better service
- Others – pay little for basic services
- Delay sensitive applications – voice, music, video, telnet
- Non-delay sensitive applications – ftp, email, newsgroups

Need for Service Differentiation

- *Same-service-to-all* model is inadequate
- A model is needed that *differentiates* packets based on their service needs.
- Fundamental approaches for service differentiation have been identified
  - *Integrated services*
  - *Differentiated services*

Integrated Services Approach

- Identifies individual packet flows between end systems based on IP addresses, port numbers and protocol field in the IP header
- Performance metrics : end-to-end delay, loss rate. Rejects new connections if resources are not available.
- Three major components
  - Admission control unit
  - Packet forwarding mechanisms
  - Resource reservation protocol (RSVP)

Problems With IntServ

- Scalability & Manageability
  - Maintaining and processing per-flow state for all flows is significantly difficult
  - Even using mechanisms like CSFQ to control flows the management and accounting of IP networks is significantly complicated
  - New application-network interfaces required
  - All networks in path should be IntServ-capable

Differentiated Services Approach

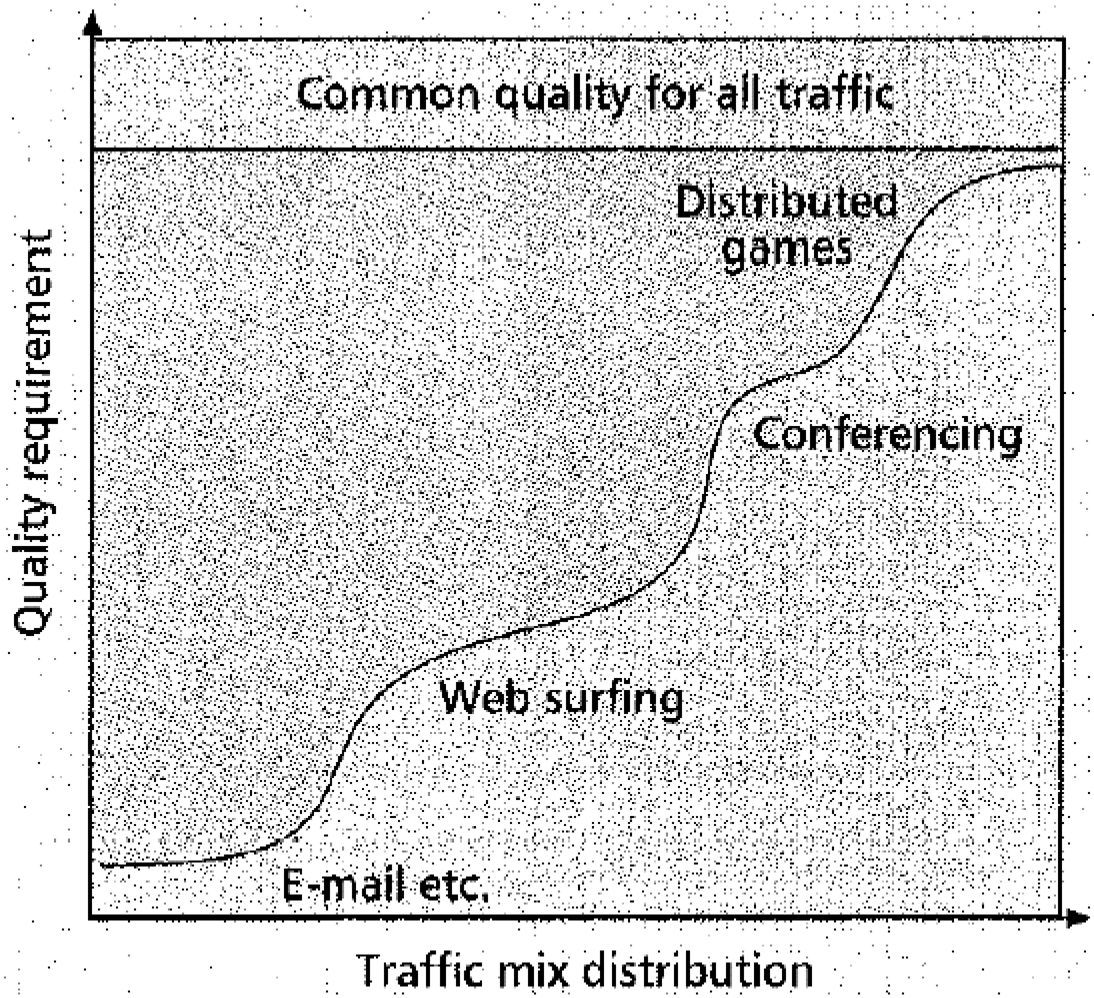
- Goal is to provide a more scalable and manageable architecture
- Two approaches
  - Absolute diffserv : absolute performance with no per-flow state information in backbone routers; Semi-static RSVP.
  - Relative diffserv
    - Similar flows aggregated into one class; Few classes
    - No absolute QoS. QoS relative between classes.

	Integrated services	Differentiated services
Granularity of service differentiation	Individual flow	Aggregate of flows
State in routers (e.g., scheduling, buffer management)	Per-flow	Per-aggregate
Traffic classification basis	Several header fields	The DS field (6 bits) of the IP header
Type of service differentiation	Deterministic or statistical guarantees	Absolute or relative assurances
Admission control	Required	Required for absolute differentiation only
Signaling protocol	Required (RSVP)	Not required for relative schemes; absolute schemes need semi-static reservations or broker agents
Coordination for service differentiation	End-to-end	Local (per-hop)
Scope of service differentiation	A unicast or multicast path	Anywhere in a network or in specific paths
Scalability	Limited by the number of flows	Limited by the number of classes of service
Network accounting	Based on flow characteristics and QoS requirement	Based on class usage
Network management	Similar to circuit-switched networks	Similar to existing IP networks
Interdomain deployment	Multilateral agreements	Bilateral agreements

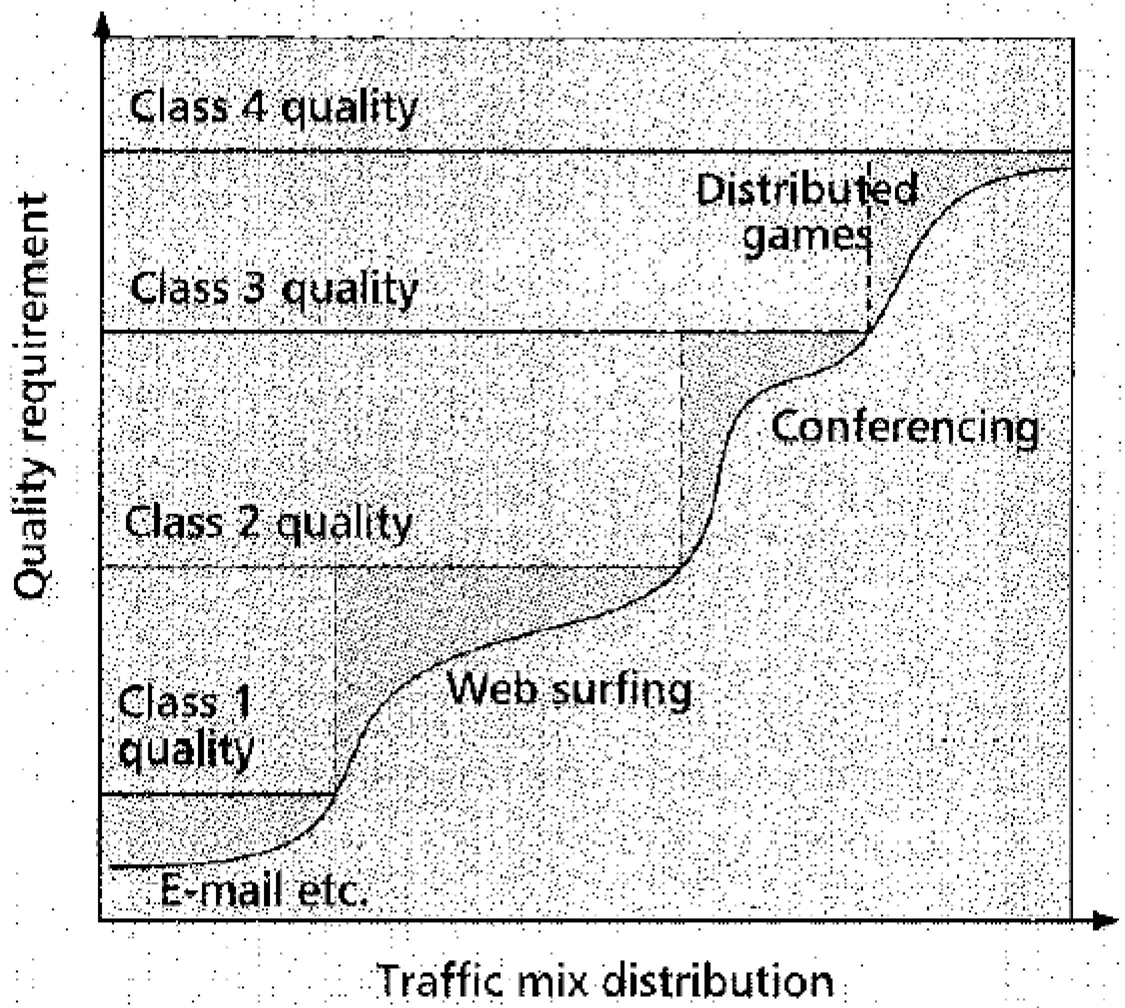
■ Table 1. *A comparison of the IntServ and DiffServ architectures.*

Fat-dumb-pipe Model

- Overprovision the network so that there is no congestion, queuing delays or losses
- Very high-capacity links relative to traffic
- Extremely inefficient in terms of network economics and resource management
- All traffic receives the same, normally very high, quality of service


Common quality for all traffic
Distributed games
Conferencing
Web surfing
E-mail etc.
Quality requirement
Traffic mix distribution

(a)


Class 4 quality
Distributed games
Class 3 quality
Conferencing
Class 2 quality
Web surfing
Class 1 quality
E-mail etc.
Quality requirement
Traffic mix distribution

(b)

Relative Differentiated Services

- Network traffic is grouped into N service classes, which are ordered based on their packet forwarding quality.
- *Class i is better (or at least no worse) than class (i-1) for $1 < i <= N$, in terms of local (per-hop) performance measures for queuing delays and packet losses.*
- The elucidation "or no worse" precludes quality degradation for higher classes over lower ones.

Relative Vs. Absolute DiffServ

- In the absolute model, an admitted user is assured of of the requested performance level. User is rejected if the required resources are not available.
- In the relative model the only assurance from the network is that a higher class will receive a better service than a lower class.

Three Relative DiffServ Models

- Strict prioritization.
  - Starvation for lower classes.
  - Not *controllable.*
- Price differentiation.
  - Rationale : higher prices lead to lower loads and thus better service quality in higher classes.
  - Ineffective in short timescales when higher classes get overloaded. Worse service quality than lower classes.
- Capacity differentiation.

Capacity Differentiation

- Allocate a larger amount of forwarding resources to higher classes, relative to the expected load in each class
- A WFQ scheduler can be configured as

$$w_i/\lambda_i > w_j/\lambda_j \quad \mathit{if}\ i > j$$

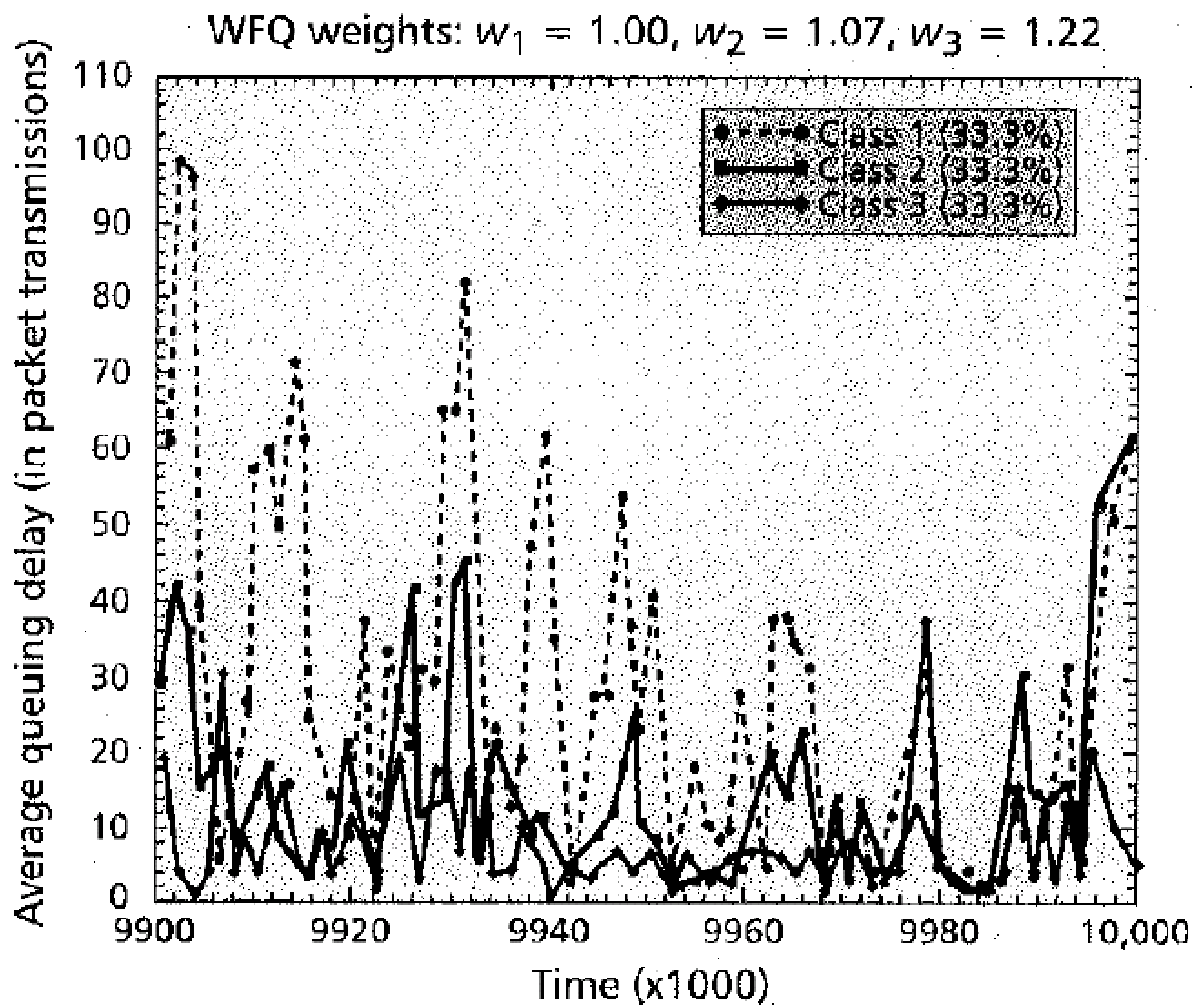
λ_i – *average arrival rate of class i*

w_i – *weight of class i*

Leading in this way to lower average delays for the higher classes

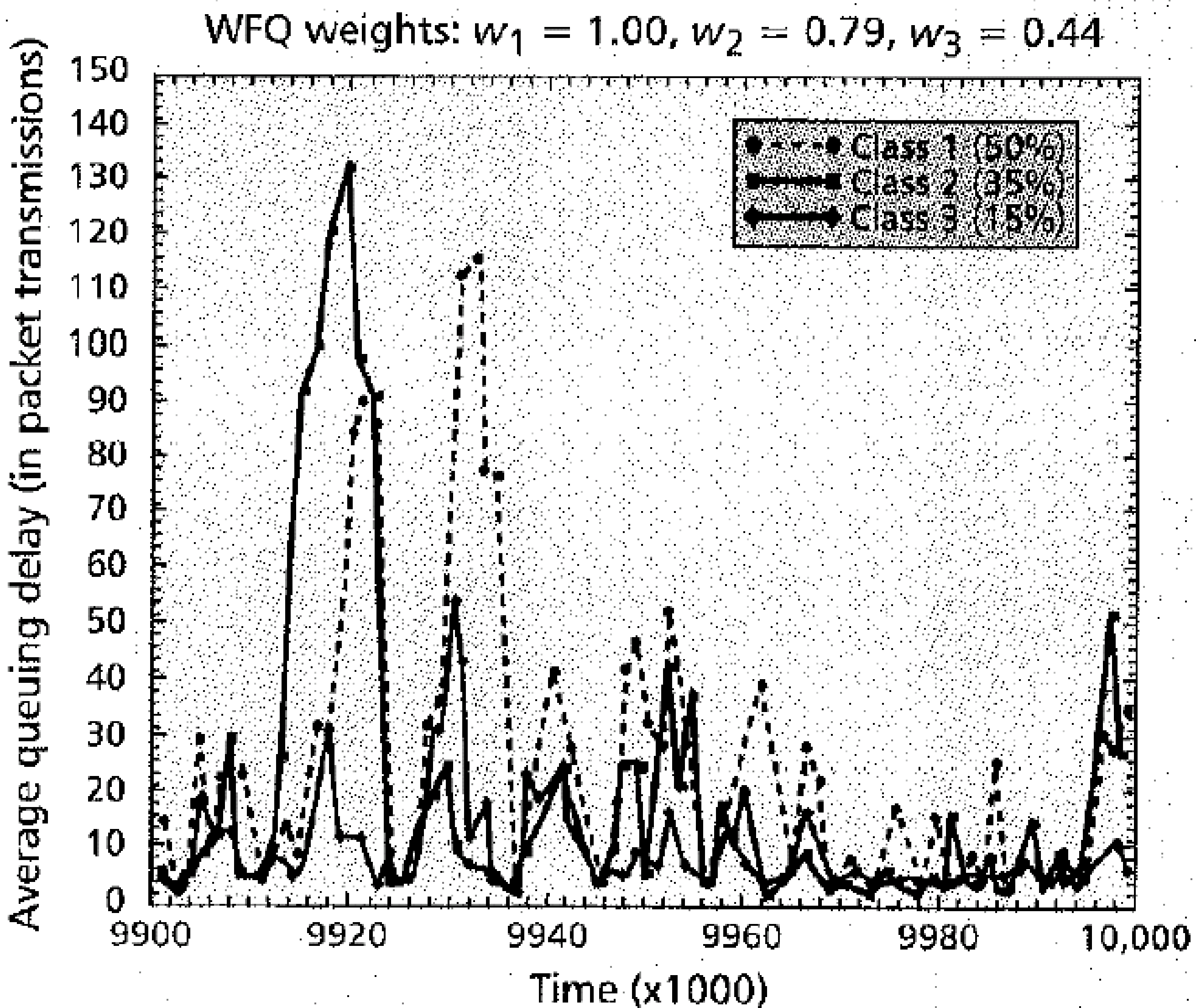
Capacity Differentiation

- An important drawback : in shorter timescales higher classes can often provide worse QoS than lower classes
- Reason : service quality depends on the short-term relation between the allocated service to a class and the arriving load in that class.
- Short-term class loads may deviate from long-term class loads over significantly large time-intervals



WFQ weights: w_1 = 1.00, w_2 = 1.07, w_3 = 1.22
Class 1 (33.3%)
Class 2 (33.3%)
Class 3 (33.3%)
Average queuing delay (in packet transmissions)
0
10
20
30
40
50
60
70
80
90
100
110
9900
9920
9940
9960
9980
10,000
Time (x1000)

(a)



WFQ weights: w1 = 1.00, w2 = 0.79, w3 = 0.44
Average queuing delay (in packet transmissions)
150
140
130
120
110
100
90
80
70
60
50
40
30
20
10
0
9900
9920
9940
9960
9980
10,000
Time (x1000)
Class 1 (50%)
Class 2 (35%)
Class 3 (15%)

(b)

Relative Service Differentiation

- Two important features desirable in a relative service differentiation model
  - Controllability – network operators should be able to adjust the quality spacing between classes based on their pricing or policy
  - Predictability – class differentiation should be consistent even in short timescales, independent of the variations of the class loads

Proportional Differentiation Model

- Rule : certain class performance metrics should be proportional to the differentiation parameters chosen by the network operator
- It is generally agreed that better network service means
  - Lower queuing delays
  - Lower likelihood of packet losses

Proportional Differentiation Model

- Suppose $q_i(t,t+\tau)$ – performance measure for class *i* in the time interval $(t,t+\tau)$, where τ is relatively small and $\tau > 0$

  $q_i(t,t+\tau)/ q_j(t,t+\tau) = c_i/c_j$

  Where $c_1 < c_2 < \ldots < c_N$ are the generic *quality differentiation parameters (QDPs)*
- Quality ratio between classes will remain fixed and controllable independent of the class loads

Proportional Delay Differentiation Model

For queuing delays $q_i(t,t+\tau) = 1/d_i\,(t,t+\tau)$ where $d_i\,(t,t+\tau)$ is the average queuing delay of the class *i* packets that departed in the time interval $(t,t+\tau)$. If there are no such packets, $d_i\,(t,t+\tau)$ is not defined.

$$d_i(t,t+\tau)/\,d_j(t,t+\tau) = \delta_i/\delta_j \qquad (1)$$

where the parameters $\{\delta_i\}$are the *delay differentiation parameters* (DDPs), being ordered as $\delta_1 > \delta_2 > \ldots > \delta_N$.

Proportional Loss Rate Differentiation Model

For loss rate $q_i(t,t+\tau) = 1/l_i\,(t,t+\tau)$ where $l_i\,(t,t+\tau)$ is the fraction of class *i* packets that were backlogged at time t or arrived during the time interval $(t,t+\tau)$, and were dropped in this same time interval.

$$l_i(t,t+\tau)/\, l_j(t,t+\tau) = \sigma_i/\sigma_j \qquad (2)$$

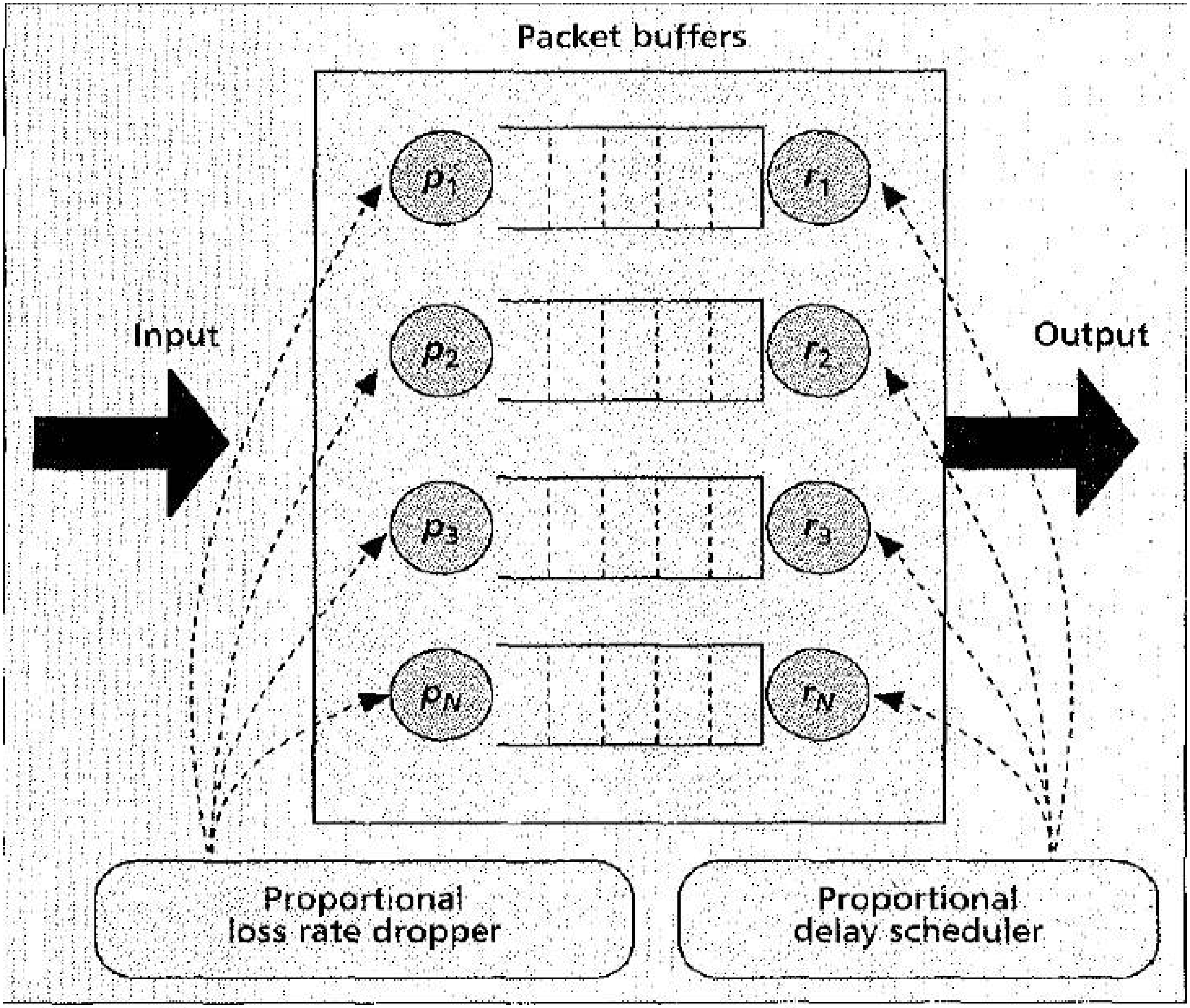
Where the parameters $\{\sigma_i\}$ are the *loss rate differentiation parameters* (LDPs), being ordered as $\sigma_1 > \sigma_2 > \ldots > \sigma_N$.

Proportional Differentiation Model

- Controllable – using QDPs
- Predictable – since τ is sufficiently small, higher classes are consistently better than lower classes even in short timescales
- Drawback - not always *feasible* using work-conserving forwarding mechanisms

	Premium service	Assured service	Proportional differentiation
Type of service	Strictly absolute	Depends on provisioning algorithm	Strictly relative
Bandwidth broker agent	Required	Not required	Not required
Route pinning	Important	Important	Not important
Operation at network ingress	Policing	Marking (IN/OUT or drop-preference)	Class-usage accounting
Endpoint adaptation	Not required	Important	Important or required

■ Table 2. *A comparison of three DiffServ models.*



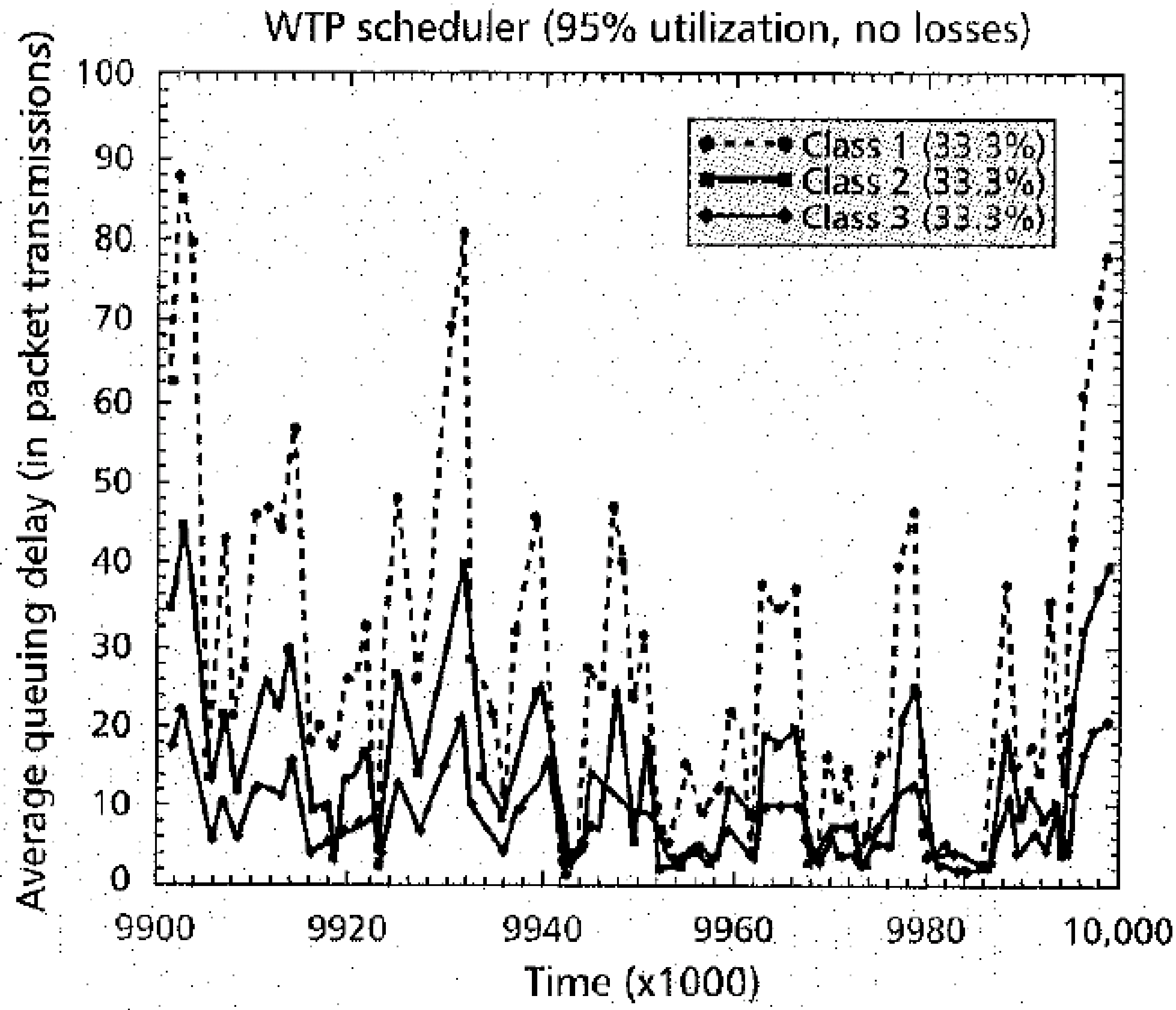
■ Figure 3. *The main components of a forwarding engine in the context of the proportional differentiation model.*

A Scheduler for Proportional Delay Differentiation

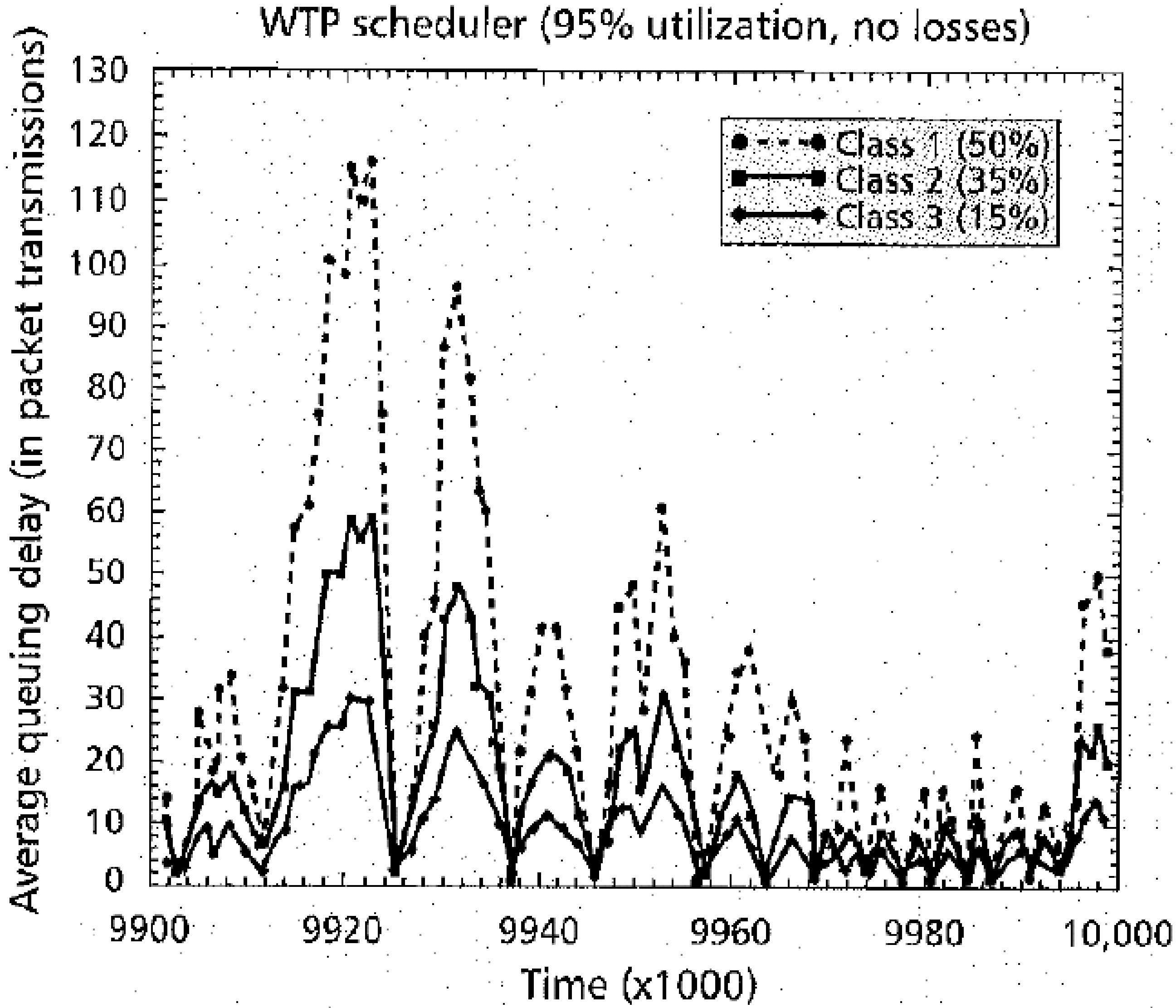
- Waiting time priority (WTP) scheduler :The priority of a packet in queue *i* at time *t* is

  $p_i(t) = w_i(t)/\delta_i$

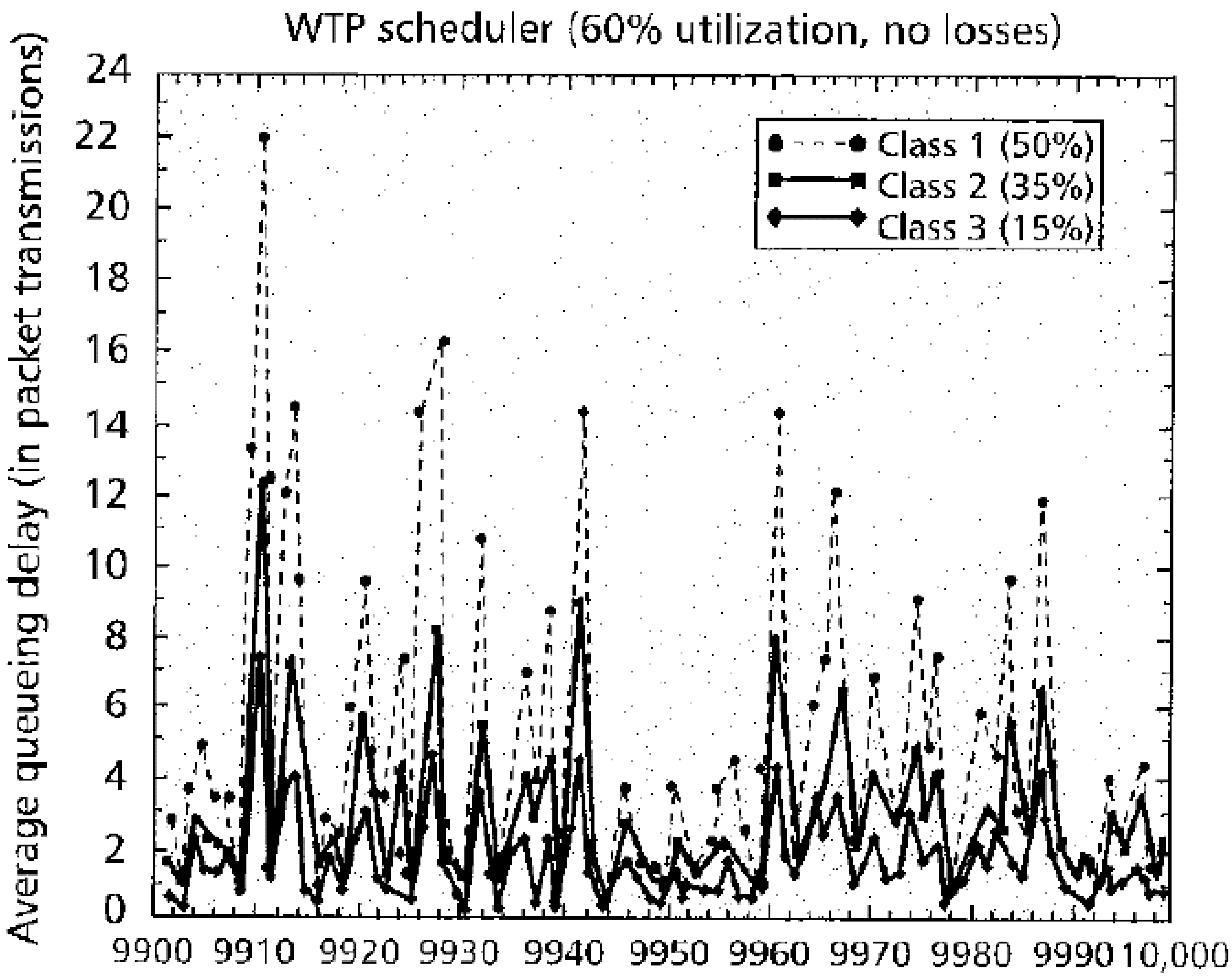
  $w_i(t)$ – waiting time of the packet at time *t*
- The WTP scheduler approximates the proportional delay differentiation model of Eq. 1 in heavy load conditions.



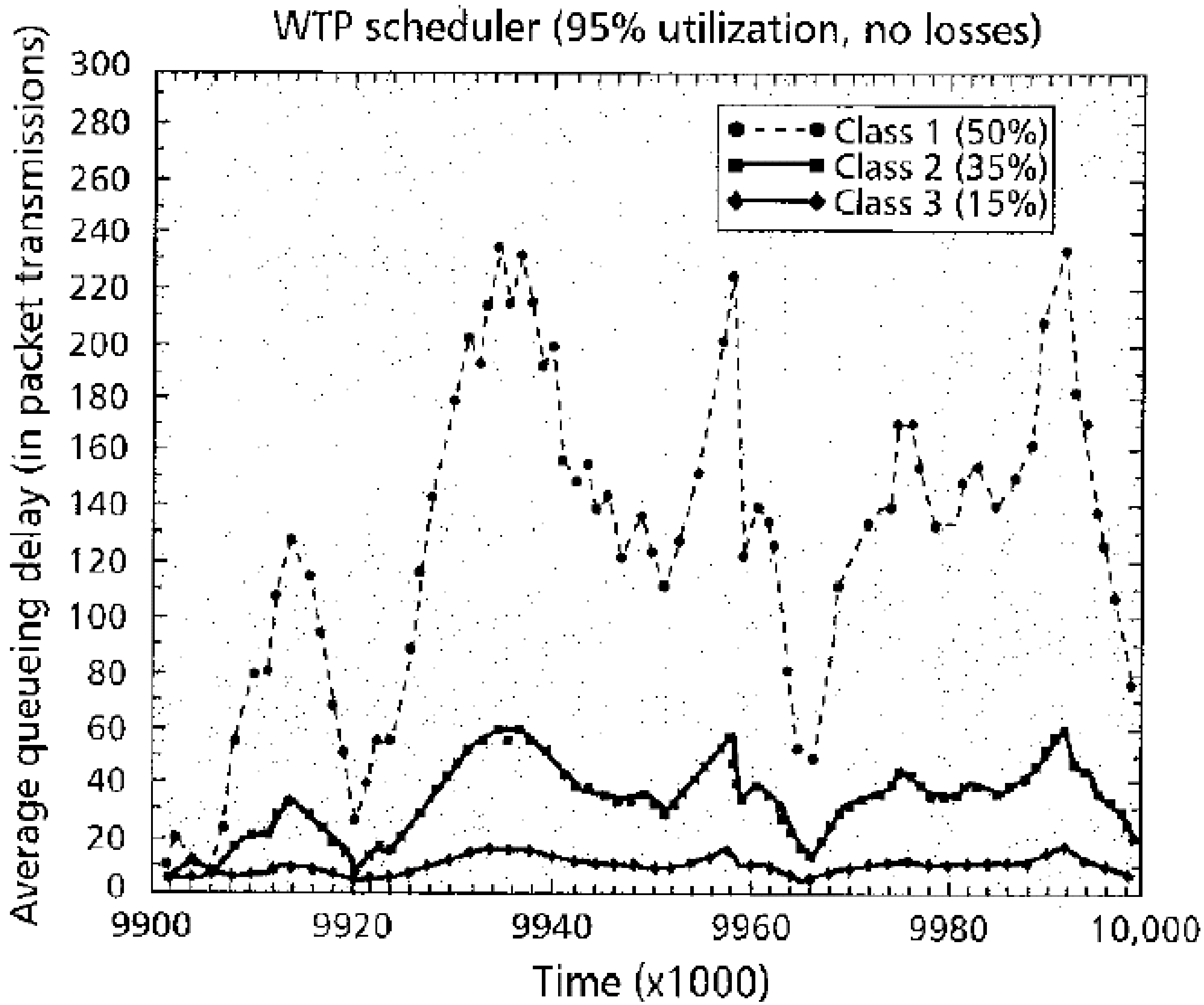
(a) Compare with Fig. 2a



(b) Compare with Fig. 2b



(a) Moderate load conditions

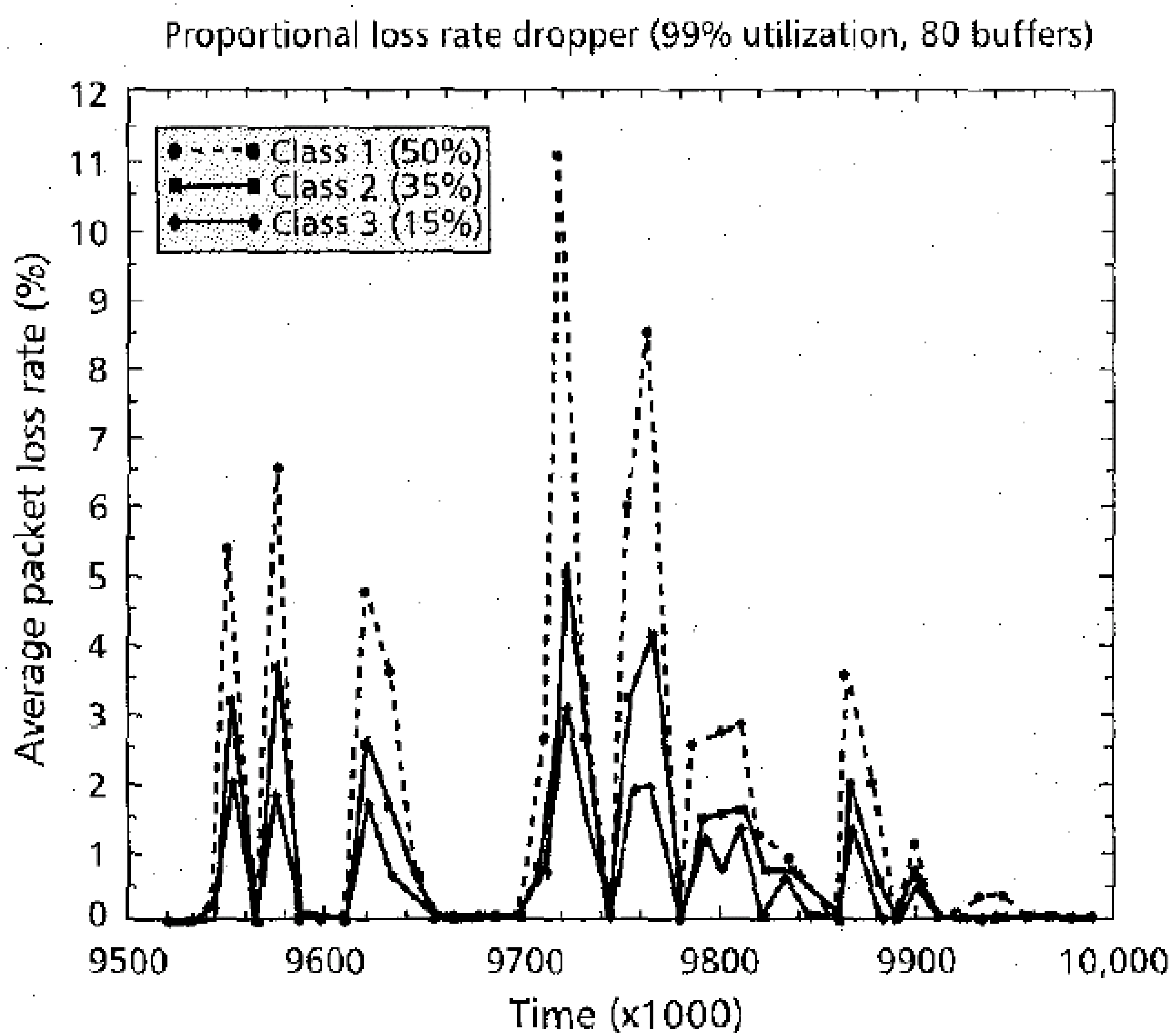


(b) Wider delay differentiation spacing

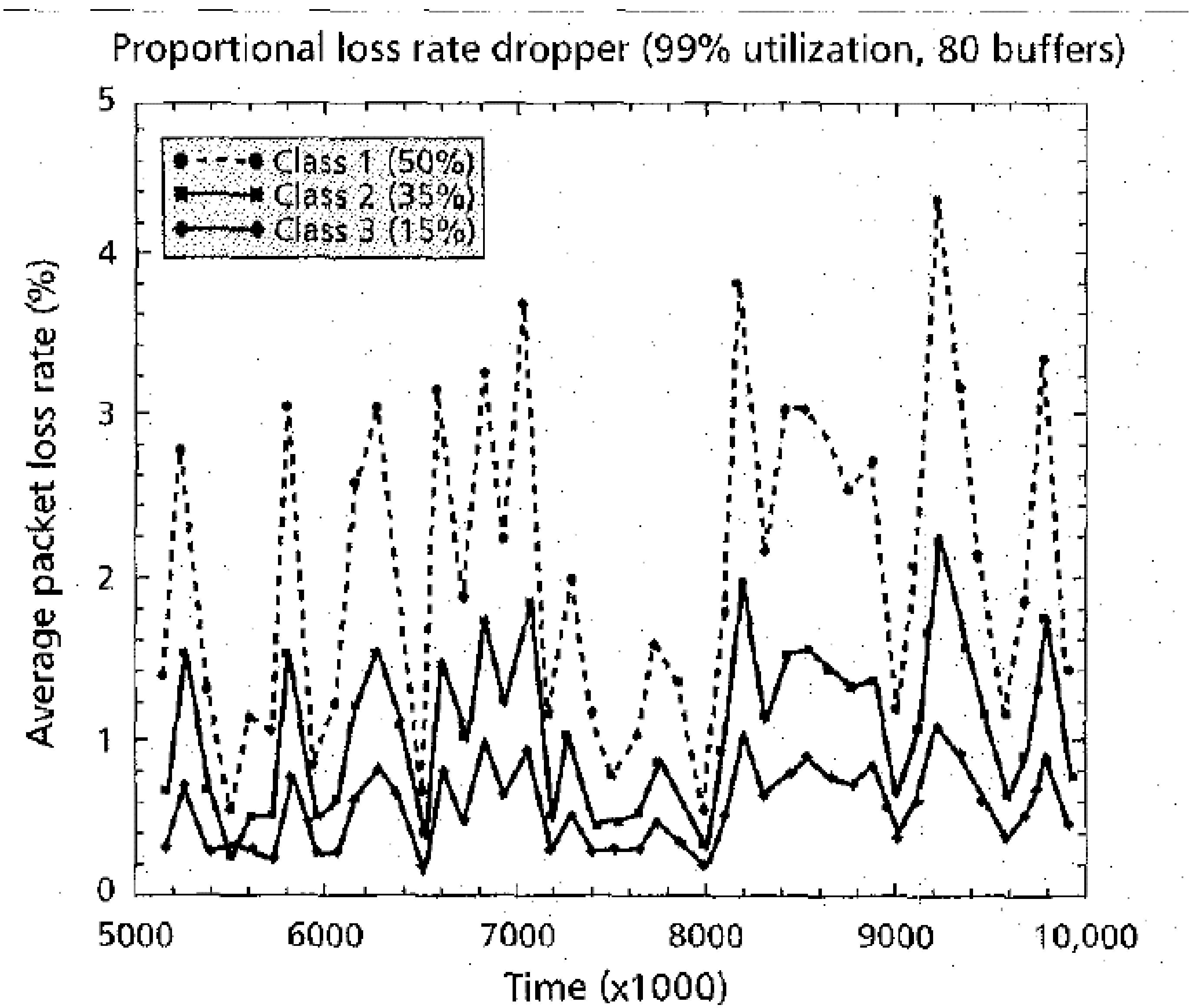
A Dropper for Proportional Loss Rate Differentiation

- The dropper maintains a *loss history buffer* (LHB), which is a cyclical queue of size K
- The dropper computes the loss rate l_i for each class i as a fraction of class i packets recorded in the LHB that were dropped.
- When a packet needs to be dropped, the dropper selects the backlogged class j with the minimum normalized loss rate; That is

  $j = argmin_i \{l_i/\sigma_i\}$.



(a) K = 1000 packets



(b) K = 10,000 packets

Conclusions

- Diffserv architecture provides services based on the taxonomy of users/applications
- Absolute diffserv – unelastic applications
- Relative diffserv
  - Users have the flexibility of selecting the forwarding class that best matches their quality-cost tradeoff
  - Easy to implement, deploy and manage

Conclusions

- Proportional diffserv
  - Allows the network operator to control the quality spacing between classes independent of class loads
  - Can provide consistent class differentiation in short timescales; Predictable
- Packet scheduling and buffer management mechanisms that approximate the behaviour of the proportional differentiation model