SCHEDULING FOR PROPORTIONAL DIFFERENTIATED SERVICES ON THE INTERNET

Manimaran Selvaraj

Master’s Thesis Presentation

Department of Electrical and Computer Engineering

Mississippi State University

Agenda

- Contributions of this work.
- Motivation.
- Problem Statement.
- QoS – a tutorial.
- Related Research.
- Our Proportional Delay Mechanism.
- Our Proportional Bandwidth Mechanism.
- Performance Evaluation and Results.
- Conclusions and Future Work.

Summary of This Work.

1. We developed
   1. a novel scheduler for delay differentiation, and
   2. a class based dropper for loss differentiation.
2. We simultaneously achieve proportional bandwidth, delay, and loss differentiation.
3. Our scheme is:
   1. Robust,
   2. Simple,
   3. Scalable, and
   4. Controls all three metrics simultaneously.

The Motivation – Why Differentiate?

- QoS measured in terms of:

  Bandwidth, Delay, and Packet Loss.
- Why delay and loss differentiation?
  - Users, who pay more, expect better service.
  - Some applications expect lower delay levels or lower loss rates.
- Why bandwidth differentiation?
  - Service providers (ISP) need to distribute bandwidth efficiently.

The Motivation – Why Differentiate all three metrics?

- More users, so many more applications.
- Today's Internet is heterogeneous. Users have different needs.
- Different levels/types of service -> different pricing levels -> good for the ISPs.
- Users want guaranteed service.

\- *So we need to control all three metrics.*

The Motivation – Why a simple, scalable, and robust solution?

- 3 independent schemes to control the 3 metrics is NOT simple.
- The Internet is huge - Solution must be scalable.
- Several new applications misbehave. They must be punished accordingly.
- Internet load fluctuates greatly. Solution must be robust.

The Problem Statement

What is available at present?

- Existing schemes control only one of the three i.e., either bandwidth, delay or loss.
- Some schemes are complex. Hence, not scalable.
- No robust solution yet. Scheme work under certain conditions only.

-*A simple, scalable and robust solution to simultaneously control all three metrics is thus needed.*

Introduction – a short tutorial on QoS – slide 1 of 3

THE INTERNET		
Past	Present	Future
▪Best-effort service model. All are EQUAL. ▪Success attributed to TCP. Past & even Present – very few UDP applications. ▪Applications *shared* resources.	▪Still best-effort only. ▪New Applications: Multimedia, voice, video, fax, online trade etc. ▪Bigger number of users than 10 years back.	▪Better than best-effort. ▪Network Capacity will be used up in future. ▪Service guarantee. ▪Predictability. ▪Measurable service. ▪Security. ▪In other words – Quality of Service (QoS) is needed.

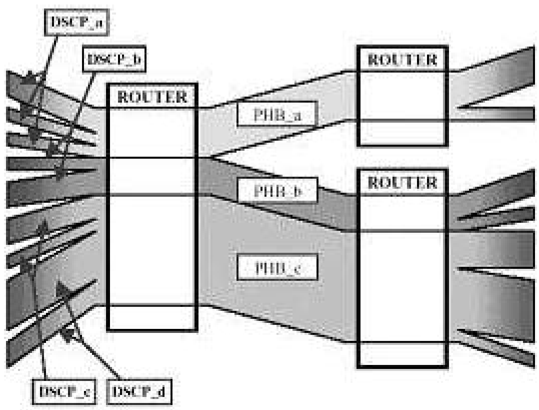
What is QoS? – slide 2 of 3

- QoS Definition: A set of rules or techniques that
  1. Help use network resources optimally to manage congestion, and
  2. Treat applications according to their needs.

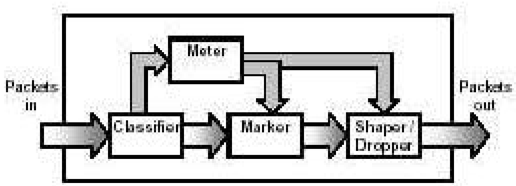
IETF's 2 QoS provisioning solutions:

- **Integrated Services**
  - Per-flow.
  - End-to-end.
  - Signalling.
  - Admission control.
  - Packet classification.
  - Packet scheduling.
- **Differentiated Services**
  - Per-aggregate.
  - No signalling.
  - Packet classification only at edges.
  - Intermediate routers read DSCP alone instead of FULL IP header.
  - Finite number of classes.

DiffServ Architecture – slide 3 of 3



DSCP to PHB mapping



Traffic conditioner at edge routers.

- Only edge routers classify packets. Others use 6 bit DSCP alone. Complexity pushed to network edge.
- DSCP selects per-hop behavior (PHB). 4 PHBs so far. *We use Assured Forwarding (AF) PHB.*
- Finite classes – so less state information at routers.
- Types: Absolute and Relative.

Related Research – slide 1 of 2

$$\frac{\bar{d}_i}{\bar{d}_j} = \frac{d_i}{d_j}$$

- First work (Dovrolis) on Proportional DiffServ contributed Proportional Average Delay (PAD), Waiting Time Priority (WTP), and Hybrid Proportional Delay (HPD) for delay differentiation.
  - PAD - selects queue with maximum normalized delay. WTP – selects queue with maximum head packet waiting time.
  - HPD – combination of PAD and WTP.
- 2 more works made WTP load adaptive. But were computationally intense.
- One work also used WFQ to achieve a proportional delay service.

Related Research – slide 2 of 2

$$\frac{l_1}{l_2} = \frac{s_1}{s_2}$$

$$\frac{BW_i}{BW_i} = \frac{b_i}{b_j}$$

- Dovrolis also proposed a counter based proportional loss differentiation scheme. Drop probability was based on the ratio of loss count value of one class with respect to another class.
- Buffer management schemes also used to proportionally loss differentiate.
- Weighted RED was used to achieve relative bandwidth service between TCP micro-flows – a per-flow experiment.

NOTE: No scheme controlled more than ONE metric.

Our Proportional Differentiation Mechanism

Adaptive HPD

- Modified Dovrolis's Hybrid Proportional Delay (HPD) algorithm.
- HPD Normalized delay is:

$$\tilde{h}_i(t) = (g)\tilde{d}_i(t) + (1-g)\tilde{w}_i(t)$$

$$\tilde{d}_i(t) = \frac{1}{d_i}\frac{S_i}{P_i}$$

$$\tilde{w}_i(t) = \frac{w_i(t)}{d_i}$$

where

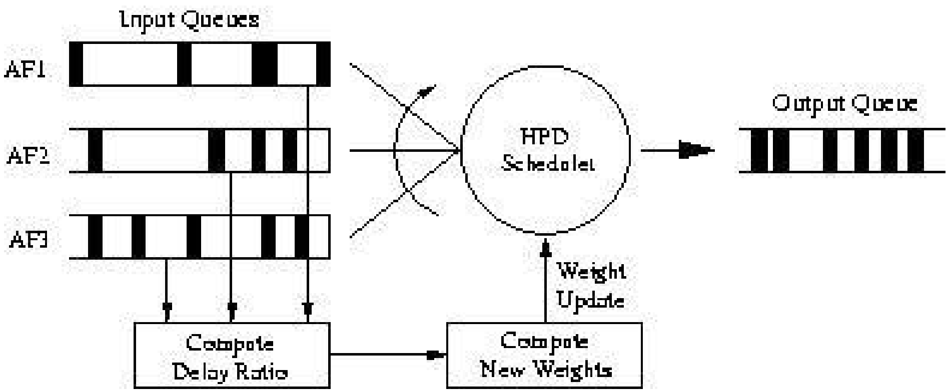
$\tilde{d}_i(t)$ is the normalized delay from PAD

$\tilde{w}_i(t)$ is the normalized wait time from WTP.

g is the HPD parameter. Decides proportion of PAD and WTP.

- Motivated by other adaptive WTP techniques.

Adaptive HPD - continued



In Short:

- Monitor delay ratio $\Delta_i = D_i / D_{i+1}$ between classes.
- Adjust weights q_i so that delay ratio is maintained.
- Decide queue to be serviced.

Adaptive HPD - continued

- D_i = average end-to-end delay experienced by each AF class.
- Delay ratio between 2 classes is measured as: $\Delta_i = D_i / D_{i+1}$
- Average delay D_i is inversely related to normalized delay $\tilde{h}_i(t)$
- So delay ratio is:

$$\Delta_i = \frac{D_i}{D_{i+1}} \approx \frac{\tilde{h}_{i+1}(t)}{\tilde{h}_i(t)}$$

Adaptive HPD - continued

- Upon each packet arrival, compute $?_i$
- Best case: $?_i = K$ (desired value)
- If $?_i$ deviates from K, adjust class weights.
- To avoid complexity: relax condition.
- Adjust weights if $?_i$ deviates outside a window (K - e, K + e).

Adaptive HPD - Initialization

Class	Initial Weight	Max Weight	Min Weight
AF1	1	1.5	0.5
AF2	2	3	1.5
AF3	4	6	3

- HPD parameter ‘g’ = 0.85
- Window e = 0.25
- Ideal delay ratio = K = 2/1 = 4/2 = 2

Adaptive HPD – The Algorithm

Assume 3 AF classes. Hence 2 delay ratios ?$_1$ and ?$_2$

1. Initialization: Set q_i, g, desired delay ratio ?$_i$. Compute initial h_i . First time, use initial ideal weights q_i.
2. When queue is served, update the parameters as follows:
   1. Calculate new ?$_2$ using $\Delta_i = \frac{D_i}{D_{i+1}} \approx \frac{\tilde{h}_{i+1}(t)}{\tilde{h}_i(t)}$
   2. Update the weights q_3 and q_2 using

$$f(q_i) = \begin{cases} q_i = q_i + \Psi \;\&\; q_{i-1} - \Psi & \rightarrow \Delta_i < K - e \\ q_i = q_i^{init} \;\&\; q_{i-1} = q_{i-1}^{init} & \rightarrow K - e < \Delta_i < K + e \\ q_i = q_i - \Phi \;\&\; q_{i-1} = q_{i-1} + \Phi & \rightarrow \Delta_i > K + e \end{cases}$$

$$\Psi = \frac{(q_{\max}^i - q_{curr}^i) x \,|\,(K - e) - \Delta_i\,|}{(q_{\max}^i - q_{\min}^i)}$$

$$\Phi = \frac{(q_{\max}^i - q_{curr}^i) x \,|\,\Delta_i - (K + e)\,|}{(q_{\max}^i - q_{\min}^i)}$$

Adaptive HPD – The Algorithm

3. Compute Calculate new ?$_1$ $\Delta_i = \frac{D_i}{D_{i+1}} \approx \frac{\tilde{h}_{i+1}(t)}{\tilde{h}_i(t)}$
4. Compute new normalized avg delay using

$$\tilde{h}_i(t) = (g)\tilde{d}_i(t) + (1-g)\tilde{w}_i(t)$$

3. Select queue to be serviced.
4. Save the updated weights for the next cycle.
5. Go back to 2.

In short: update weights of 2 highest classes. Then update the weight of the lower priority class based on its predecessor.

Bandwidth Differentiation

- Delay and Loss differentiation results in Bandwidth differentiation.
- Based on Random Early Detection (RED).
- Drops packets randomly, before actual congestion can take place.
- Parameters are:
  - Max_{th} – maximum drop threshold
  - Min_{th} – minimum drop threshold
  - Max_{p} - maximum drop probability
- In short, estimate average queue length AQS.
  - – Start dropping packets with low probability when AQS = Min_{th}
  - – The larger the AQS, the more likely a packet is dropped.
  - – All packets are dropped when AQS >= Max_{th}

Colored RED

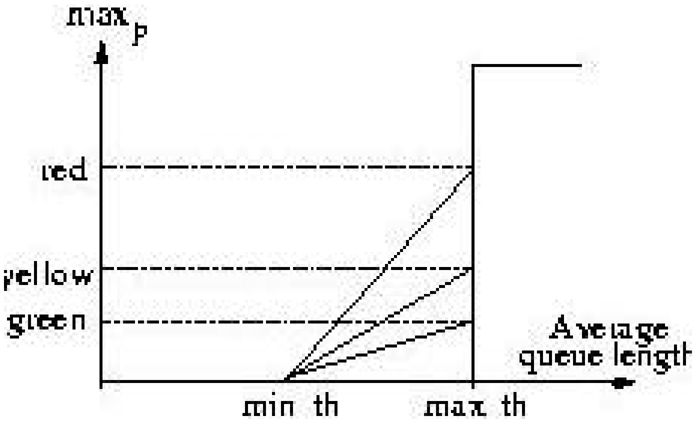
- A version of multi-class RED.
- Colored RED – a RED set for each class/color.
- We used 3 classes. So 3 colors – red, yellow, and green.
- Similar to WRED, but multi-class RED was never tried for per-class differentiation.
- All packets of one AF class are painted (header marked) the same.
- Inside a class/color – 2 sets of drop thresholds.

Colored RED - Parameters

- Either max_{th}, min_{th} or max_p can be set proportionally for different classes. In CRED max_p alone proportional.

$$\frac{\text{RED Drop probability of class i}}{\text{RED Drop probability of class j}} = \frac{s_i}{s_j}$$

- In CRED only max_p is proportional.



Only maxp is proportional

Queue Type	min_{th}	max_{th}	max_p
Red/ AF11	20	40	0.08
Red/AF12	10	20	0.16
Yellow/AF21	20	40	0.04
Yellow/AF22	10	20	0.08
Green/AF31	20	40	0.04
Green/AF32	10	20	0.02

Colored RED – AQS Calculation

- Differentiation further enhanced by average queue size calculation.
  - Independent queue size – one AQS for each class.
  - Total queue size – Overall AQS for all classes.
  - Additive queue size – cumulative AQS.
- Average Queue Size
  - Decides the packet's fate.
  - In most RED versions, AQS of one class combined with another class.
  - In CRED,
    AQS_{AFi} = TSW estimate based on AFi packets alone.
  - This adheres to AF PHB specs.

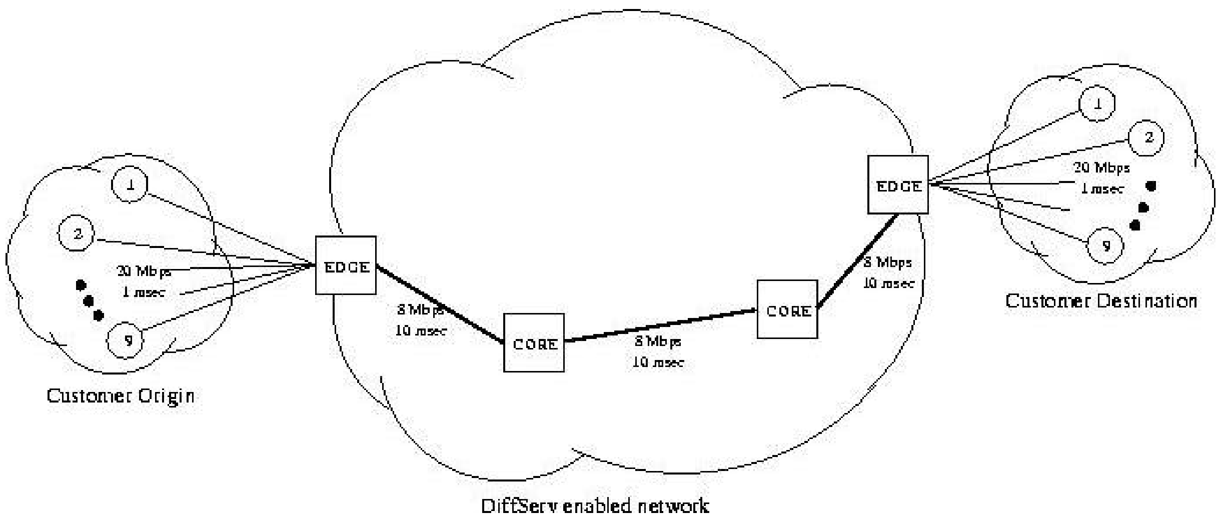
CRED – The Algorithm

1. As packet arrives, check color.
2. Compute the corresponding queue's AQS (independent computation)
3. Apply CRED with corresponding RED parameters.
4. Decide packet's fate.

Performance Evaluation

- Simulation Tool: LBNL's network simulator ns-2.
- Several traffic sources were used for testing:
  - FTP, CBR, On-/Off- exponential and Pareto sources, and flows with different RTTs.
- TCP and UDP was used to study their interactions.
- Compared with original HPD and RIO combination.

Simulation Model

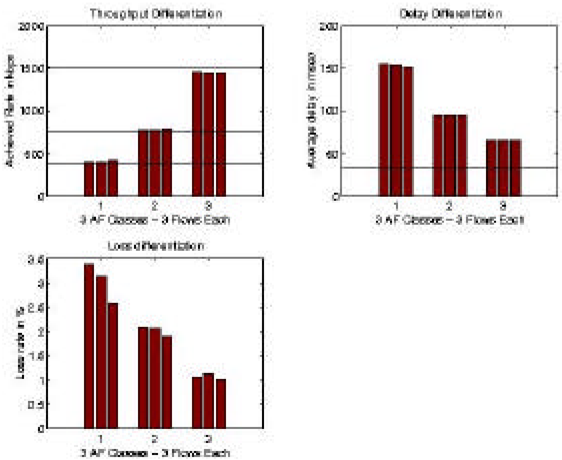


- Adaptive HPD and CRED are implemented at the edge and core routers.
- 9 flows (3 for each class). Classes have proportional bandwidth, delay, and loss requirements.

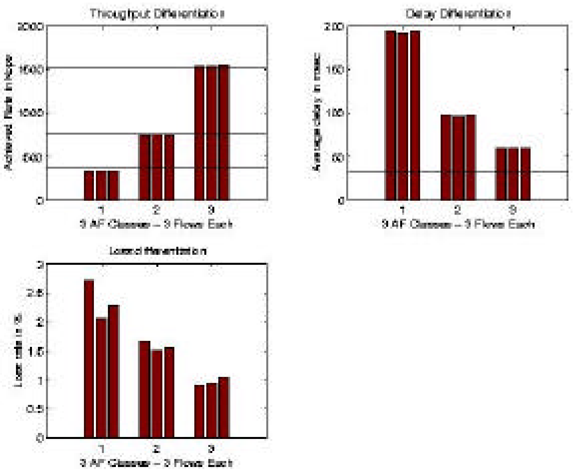
Measurements Made:

- Average throughput.
- Average one way end-to-end delay.
- Packet loss rate.
- Delay ratio between classes.
- Bandwidth ratio.
- Fairness index.

Results – 1 (FTP sources)



Original HPD and RIO

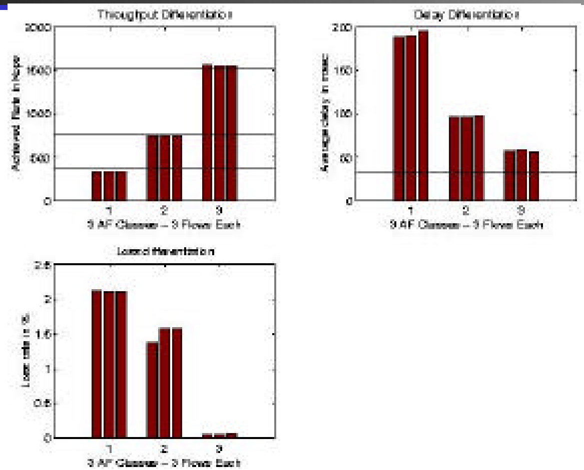


Adaptive HPD and CRED

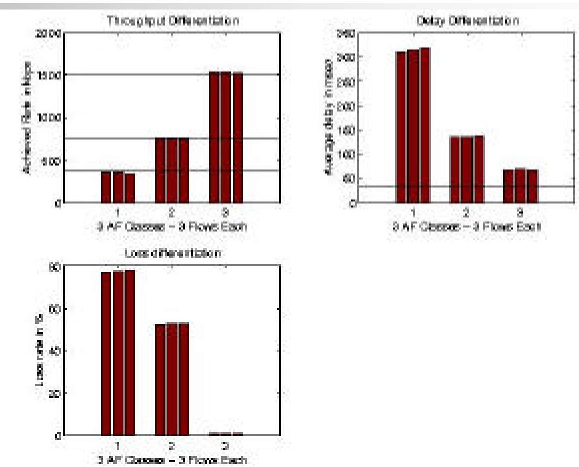
Scheme	Average Class Delay in msec			Delay Ratio	
	AF1	AF2	AF3	AF1/ AF2	AF2/ AF3
AHPD & CRED	193.67	97.4	59.66	1.99	1.63
HPD & RIO	153.6	94.9	65.53	1.62	1.45

Delay Ratio Comparison

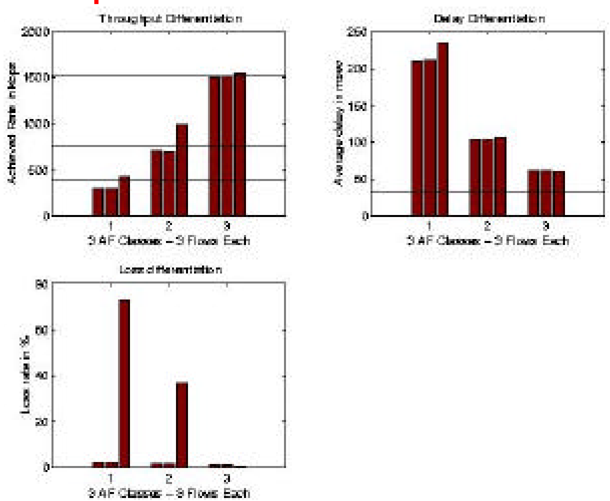
Results – 2 (CBR sources)



Hpd1-cbr1



Hpd1-cbr4

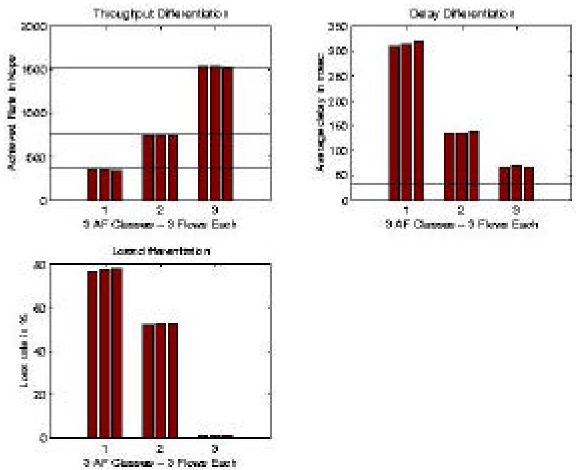


Hpd1-cbr2: F.I = 0.96, 0.97 and 0.99

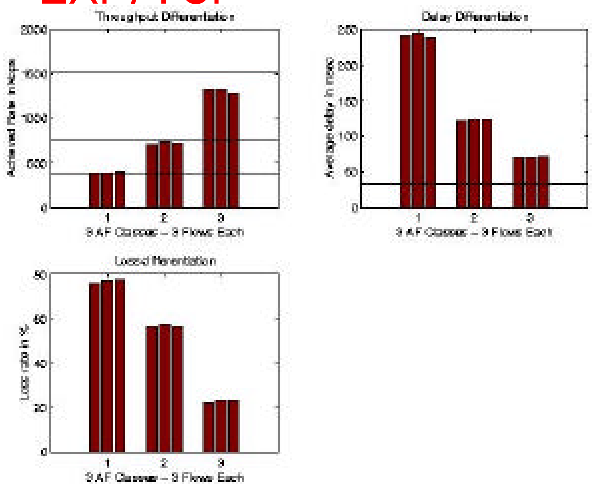
Traffic Types	Scheme	Average Class Delay			Delay Ratio	
		AF1	AF2	AF3	AF1/ AF2	AF2/ AF3
All flows are CBR/TCP	AHPD & CRED	191.65	96.90	56.83	1.97	1.7
	HPD & RIO	152.55	94.47	65.36	1.61	1.45
One flow in each class is CBR/UDP	AHPD & CRED	218.33	104.43	60.92	2.09	1.71
	HPD & RIO	170.48	103.24	69.67	1.65	1.48
All flows are CBR/UDP	AHPD & CRED	314.78	136.53	68.29	2.31	1.99
	HPD & RIO	216.49	126.32	81.13	1.71	1.56

Delay Ratio Comparison

Results –3 (Exponential Sources)



EXP/TCP

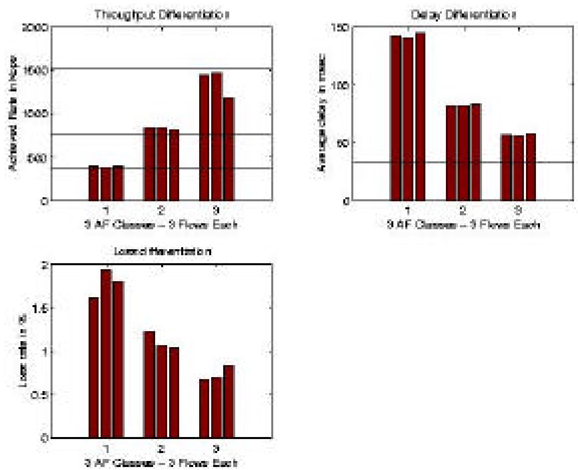


EXP/UDP

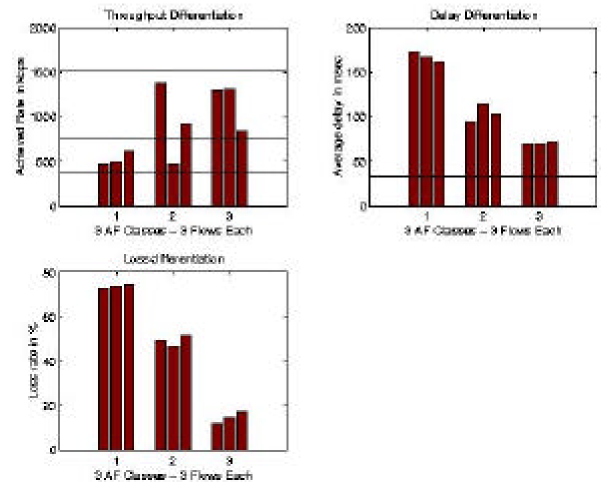
Traffic Type	Scheme	Average class delay			Delay Ratio	
		AF1	AF2	AF3	AF1/ AF2	AF2/ AF3
Exponential over TCP	AHPD & CRED	186.64	94.94	58.66	1.97	1.61
	HPD & RIO	151.83	94.07	65.10	1.61	1.45
Exponential over UDP	AHPD &CRED	241.61	123.46	70.49	1.96	1.75
	HPD &RIO	217.43	128.93	82.38	1.68	1.56

Delay ratio comparison

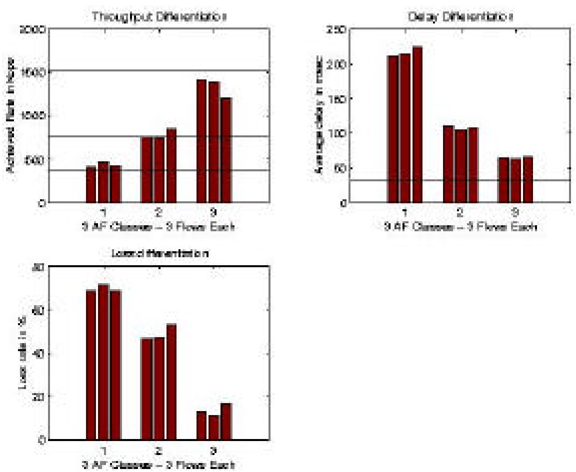
Results – 4 (Pareto Sources)



AHPD & CRED – Pareto over TCP



HPD & RIO – Pareto over UDP

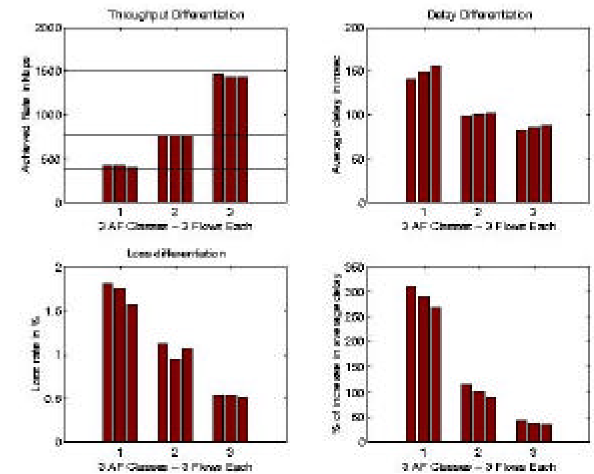


AHPD & CRED – Pareto over UDP

Traffic Type	Scheme	Average class delay			Delay Ratio	
		AF1	AF2	AF3	AF1/ AF2	AF2/ AF3
Exponential over TCP	AHPD & CRED	143.33	82.59	56.93	1.73	1.45
	HPD & RIO	135.96	86.39	61.40	1.57	1.41
Exponential over UDP	AHPD & CRED	181.27	97.22	63.12	1.86	1.54
	HPD &RIO	167.46	104.46	69.96	1.60	1.49

Delay Ratio Comparison

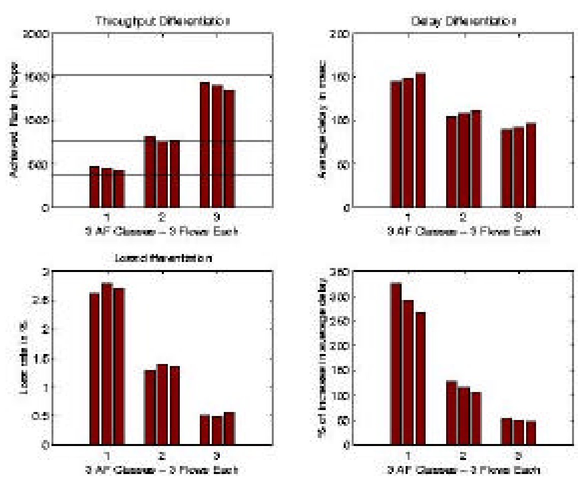
Results 5 – (Flows differing RTT)



AHPD & CRED

Scheme	Average Class Delay in msec			Delay Ratio	
	AF1	AF2	AF3	AF1/ AF2	AF2/ AF3
AHPD & CRED	147.97	100.50	85.90	1.47	1.17
HPD & RIO	149.02	107.87	92.93	1.38	1.16

Delay Ratio Comparison



HPD & RIO

Scheme	Average Class Bandwidth in kbps			Bandwidth Ratio	
	AF1	AF2	AF3	AF1/ AF2	AF2/ AF3
AHPD & CRED	415.12	761.11	1433.90	1.83	1.89
HPD & RIO	442.43	782.98	1397.22	1.77	1.78

Bandwidth Ratio Comparison

Discussions on Results

- RED parameters have a big influence on performance.
- Assign delay tolerant applications to low priority classes.
- Sources using UDP lose almost all packets in excess of agreement.
- Delay differentiation good even in the presence of UDP.
- Tests with Pareto traffic show that our scheme is robust.
- Our scheme is also tolerant to variation in RTT.

Conclusions

- We presented:
  - a scheduler to control **delay**
  - a class based dropper to control **loss**
  - The combination results in simultaneous proportional bandwidth, delay, and loss.
- Highlight of our scheme:
  - Simple, unified, robust, and above all, controls all three QoS metrics.

Future Work

- CRED can be further improved by maintaining a history of packet loss. This packet loss history can be used to determine the packet's fate.
- DiffServ over Multiprotocol Label Switching (MPLS) enabled network.