An Adaptive Cross-Layer Scheduler for Improved QoS Support of Multiclass Data Services on Wireless Systems

IEEE JOURNAL ON
SELECTED AREA IN COMMUNICATIONS,
VOL. 23, NO. 2, FEBRUARY 2005

Presented by Tsung-Yuan Hsu

Outline

- Introduction
- Commonly Used Scheduling Algorithm
- Adaptive Cross-Layer (ACL) Scheduling Algorithm
- Performance Measures
- Simulation
- Conclusion

Introduction

- Two major focuses on QoS in wireless systems
  1. Minimizing packet delay
  2. Maximizing user throughput

- Satisfying one measure sacrifices the other.

- This paper introduces a new packet scheduler which minimizes a prescribed cost function given the current channel qualities and delay states of the packets in the queue.

Commonly Used Scheduling Algorithm
a) Weighted Fair Queue (WFQ)

- The WFQ scheduling algorithm separates packets into queues according to their delay class.
- These queues are then serviced in weighted round-robin fashion.
- The weights are based on the relative performance requirements among delay classes.
- The scheduling decision does not factor in packet delays or channel qualities directly.
- The WFQ performs poorly with respect to delay and throughput measures under medium to heavy load conditions.

Commonly Used Scheduling Algorithm b) Earliest Deadline First (EDF)

- It queues packets in ascending order according to packet deadlines, then schedules the first packet in the queue.
- The EDF algorithm does not take channel quality into account.
- When the system load is heavy and interference level is high, the EDF scheduler stalls on low throughput channels resulting in further increases in load and interference.

Compare between WFQ and EDF

load conditions / compare	medium	heavy
better	EDF	WFQ
reason	EDF algorithm schedules based on delay deadlines, it outperforms the WFQ algorithm with respect to delay under medium load conditions	WFQ algorithm avoids this phenomenon by circulating across all user/channel combinations in the queue

Adaptive Cross-Layer Scheduling Algorithm

- The name derives from the fact that the algorithm adapts the scheduling order to changes in variables across layers
  1. packet delay deadlines on the link layer
  2. channel qualities on the physical layer

Adaptive Cross-Layer Scheduling Algorithm

- ACL algorithm schedules packets in the order that minimizes the cost function J

$$J(v)=\sum_{i=1}^{M}\left[(1-\beta)\frac{\hat{d}_i}{r_i}+\beta \max\left\{0, \frac{\hat{d}_i}{r_i}-1\right\}^{\gamma}\right]$$

  - v : a permutation of scheduling order
  - M : the total number of packets in the queue
  - $\hat{d}_i$: the delay estimate
  - r_i : the delay requirement
  - β : a weighting parameter between the estimated normalized packet delays and missed deadline penalties
  - γ : determines the relative cost of incremental delays beyond the packet deadline

Adaptive Cross-Layer Scheduling Algorithm

$$\hat{d}_i = c_i + \sum_{j=1}^{i} \frac{b_j}{e_j}$$

- c_i : current delay
- b_j : the packet's remaining size
- e_j : estimated channel bit-rate

$$r(c,l) = \begin{cases} \frac{0.5l}{128}, & c=1, l \le 128 \\ 0.5 + \frac{1.5(l-128)}{1024-128}, & c=1, l > 128 \\ \frac{5l}{128}, & c=2, l \le 128 \\ 5 + \frac{10(l-128)}{1024-128}, & c=2, l > 128 \\ \frac{50l}{128} & c=3, l \le 128 \\ 50 + \frac{25(l-128)}{1024-128} & c=3, l > 128 \end{cases}$$

DELAY REQUIREMENTS FOR DIFFERENT PRIORITY CLASSES AND PACKET SIZES

Priority Class	Packet Size & Delay Requirement			
	128 octets		1024 octets	
	Mean (s)	95% (s)	Mean (s)	95% (s)
1. Predictive	0.5	1.5	2	7
2. Predictive	5	25	15	75
3. Predictive	50	250	75	375
4. Best Effort	Unspecified			

Adaptive Cross-Layer Scheduling Algorithm

- ACL algorithm will reorder the packets in the queue according to the permutation that minimizes J in three conditions
  1. a new packet enters the queue
  2. an existing packet leaves the queue due to dropping or handover
  3. a queued packet's channel quality changes

Adaptive Cross-Layer Scheduling Algorithm

- In a system with high mobility and bursty data traffic, channel qualities change at a rapid rate, determining the full scheduling order at any one time is wasteful.
- As user mobility increases, either causes a drop in scheduler performance or increases the amount of computation.
- As traffic conditions become more bursty, such situations call for a simplified version of the ACL scheduling algorithm.

Simplified ACL Scheduling Algorithm

- Instead, at any given transmission opportunity, we prefer to determine the “packet-to-send”.
- Determine the relative scheduling order of any two packets in the queue without knowing the entire order.
- Setting β =0 $\frac{\hat{d}_i}{r_i} \rightarrow b_j/e_j r_i$
- Instead of calculating the cost of two entire scheduling orders, we only need to calculate two simple ratios.
- Tradeoff between accuracy and processing speed.

Simplified ACL Scheduling Algorithm

- When $\beta = 0$, the ACL simplifies to the following steps
  1. When a new packet arrives to the queue, we place it at the end
  2. At each scheduling event, we determine the index, of the ”packet-to-send” in the original queue via the following pseudocode:

```
set i = 1
for j = 2 : M
  if b_j/e_j r_i < b_i/e_i r_j
  then set i = j.
end
```

Performance Measure

- Determine the system's packet delay performance
  1. NPD (average normalized packet delay)
  2. PEN (missed packet deadline penalty)

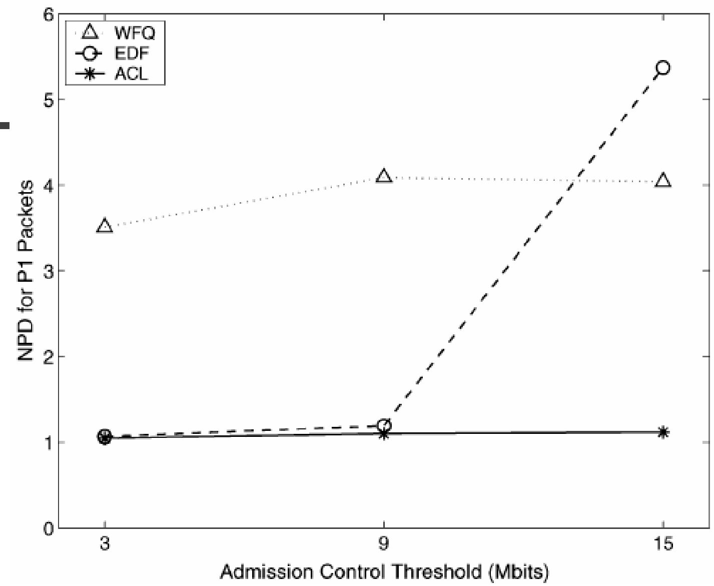
- Determine user throughput
  1. EUT (average effective user throughput)

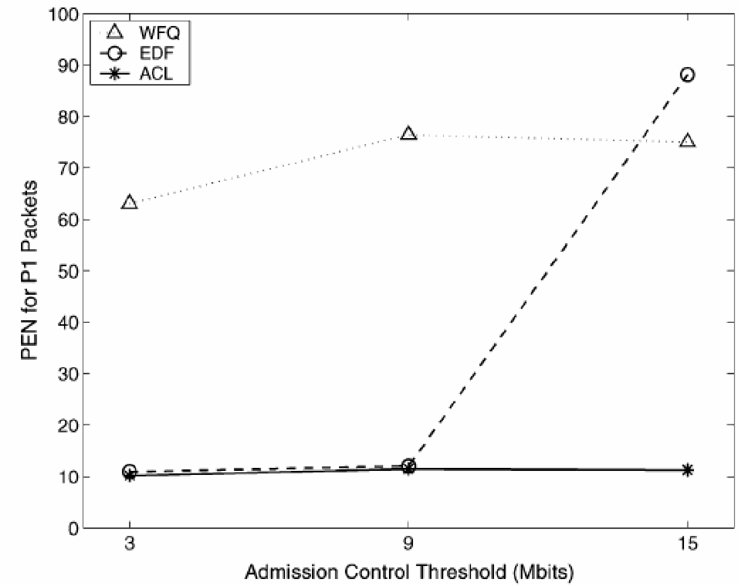
- Determine the load that a scheduler can support
  1. Percent blocking
  2. Percent dropping

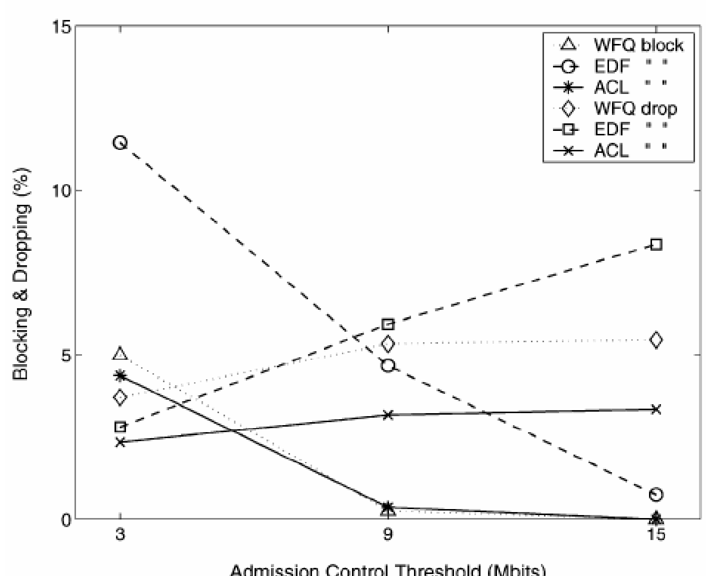
SIMULATION MODELS AND PARAMETERS

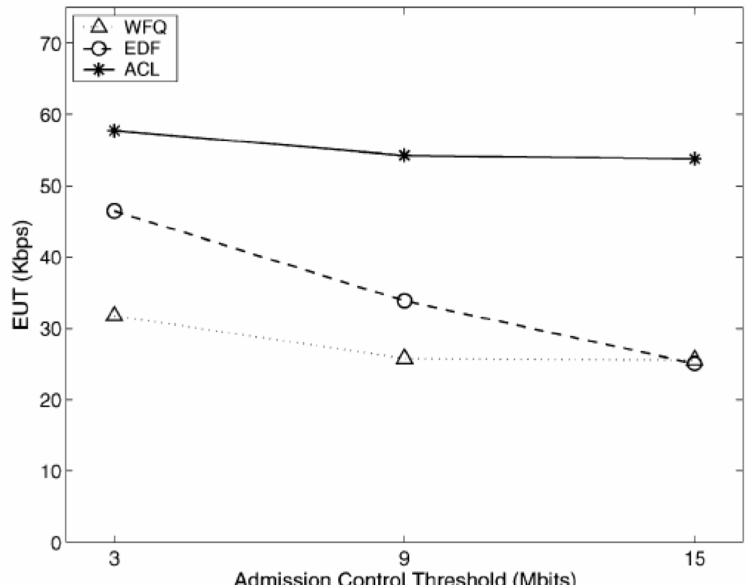
Traffic Environment	
User session inter-arrival time	Exponential ($\lambda = 4.35$ sessions/s)
Files per user session	Geometric ($\mu = 10$ files)
File inter-arrival time	Pareto ($\mu = 10$ s)
File types	Email (Priority 2) or WWW (Priority 1)
Traffic scenarios	Percent Email vs. WWW: (1) 50/50, (2) 75/25, (3) 25/75
Email file size	Modified Cauchy ($\mu = 4$ Kbytes)
WWW file size	Log-normal ($\mu = 4.1$ Kbytes, $\sigma = 44$ Kbytes)
Physical Environment	
Path-loss	Exponential (d^{α}, $\alpha = 4.5$)
Large scale fading	Log-normal ($\sigma = 10$ dB), correlation distance = 110 m
Small scale fading	Rayleigh
Channel response	GSM Typical Urban (TU3), downlink
Mobile speed	3 km/hr
Frequency reuse (Site/Sector)	$1/1$
Cell radius	300 m
System Environment	
Modulation/coding	MCS9, Incremental Redundancy
Diversity	No frequency hopping, antenna diversity gain = 3 dB
Power control	Mode-0 only, shutoff threshold = 1 dB, transmit power = 20 W
Admission control	Blocking threshold = $3, 9, 15$, or 27 Mbits
Dropping	Leaky bucket filter, bucket size $B = 280$ or 560 tokens
Mobile types	Multi-slot (4)
Simulation Time	
Time step	1 RLC block (20 ms)
Simulation length	$240,000$ RLC blocks

Leaky bucket 280,
50% e-mail and 50% WWW traffic

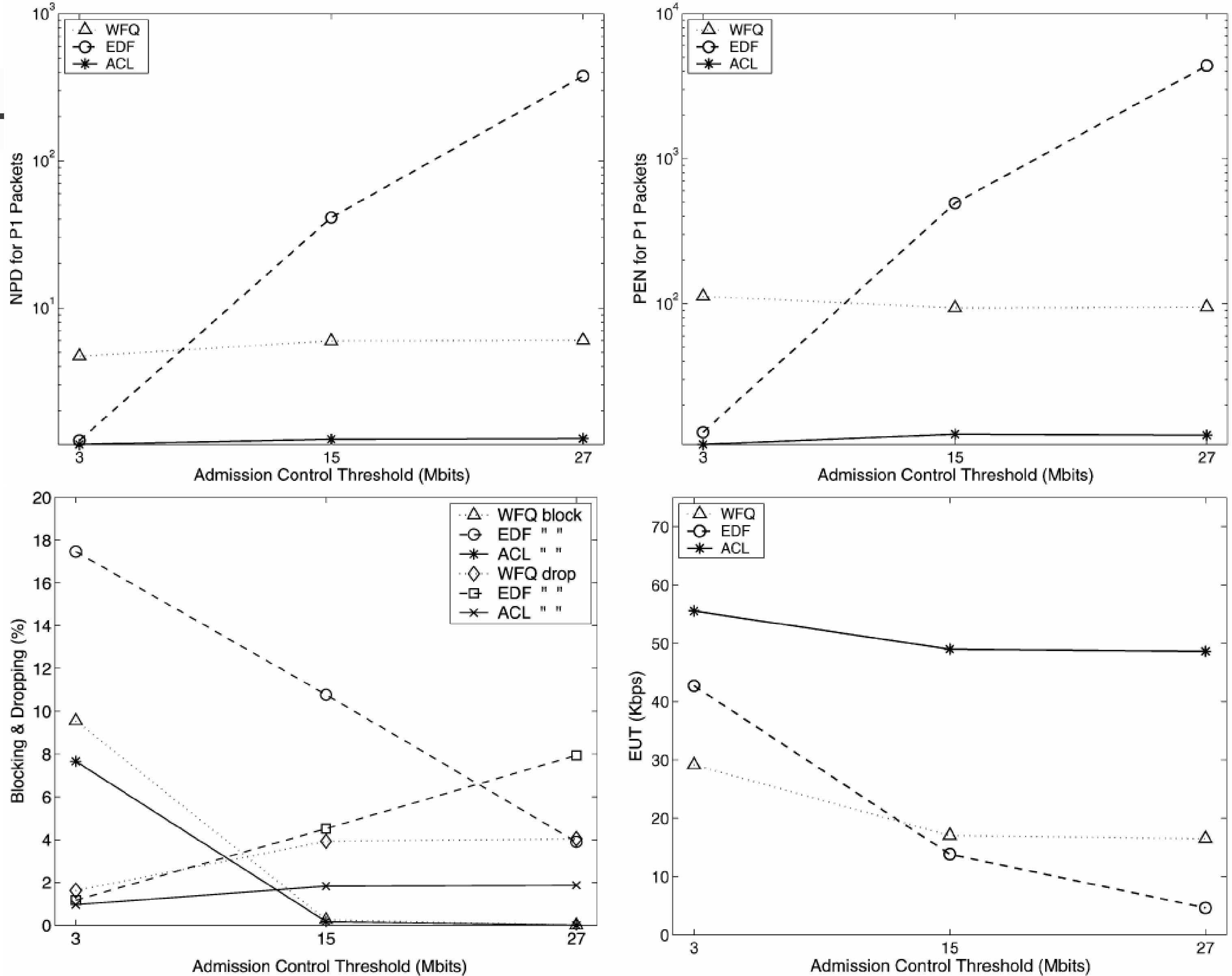




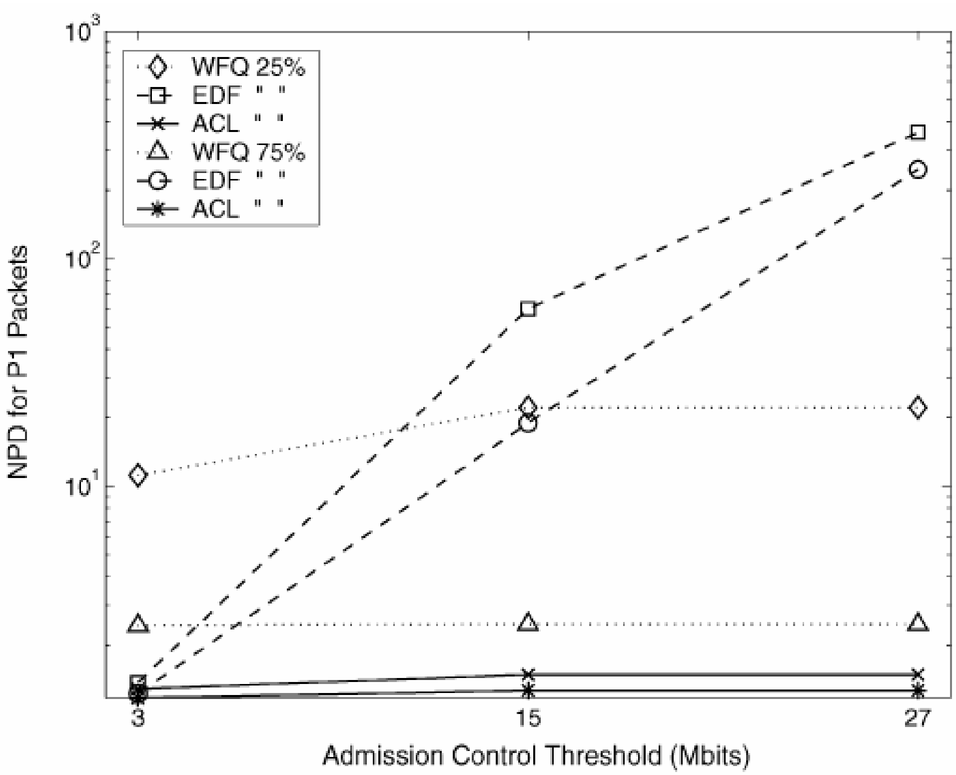


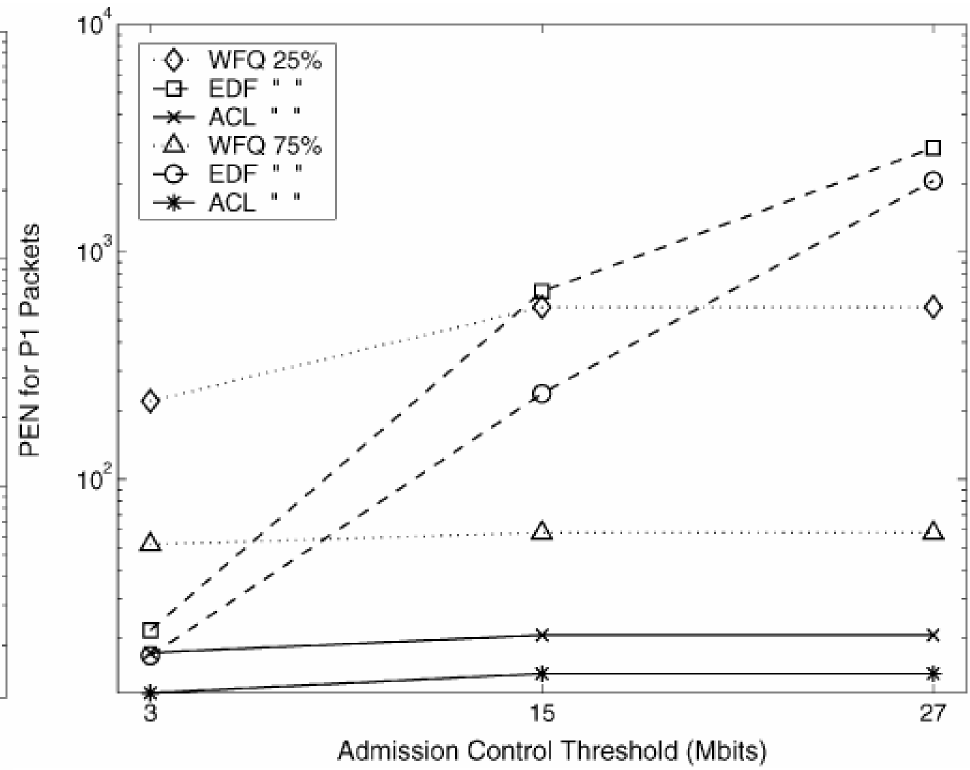


Leaky bucket 560, 50% e-mail and 50% WWW traffic



traffic scenario 2 (75% e-mail)
traffic scenario 3 (25% e-mail)





Conclusion

- Simulations confirm that the ACL scheduling algorithm greatly outperforms both the WFQ and EDF schedulers with respect to
  1. average normalized packet delay
  2. missed packet deadline penalty
  3. average effective user throughput
  4. user blocking
  5. user dropping