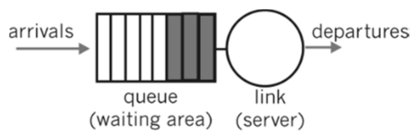
How are resources shared? Scheduling

Outline

- Sharing Router Output Port
  - What is packet scheduling?
  - Why is it needed?
  - What are the requirements for scheduling algorithms?
  - Specific algorithms
    - FIFO (done)
    - Non-preemptive priority (done)
    - Round Robin (RR)
    - Weighted Fair Queuing (WFQ)
    - Proportional Fair (PFQ)
- How scheduling is used in access networks

What is packet scheduling?

- If there is a backlog of packets to send, scheduling selects the next packet to get service
- No-scheduling-FIFO

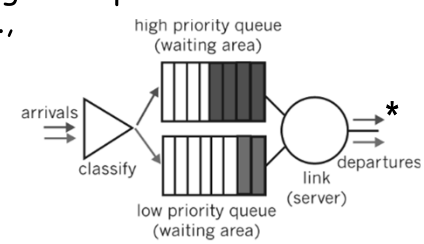


*

* From: *Computer Networking: A Top Down Approach Featuring the Internet*, 2nd edition. Jim Kurose, Keith Ross Addison-Wesley, July 2002.

What is packet scheduling?

- More interesting when packets are "different", e.g.,
  - Class
  - Urgency



*

- The server decides which packet to send next
- A scheduling algorithm is used to make the decision

* From: *Computer Networking: A Top Down Approach Featuring the Internet*, 2nd edition. Jim Kurose, Keith Ross Addison-Wesley, July 2002.

Why is it needed?

- Decides who is next.
- Need fairness, prevent one user from getting all the service
- Some packets have deadlines, e.g., for real-time services
- Need scheduling to provide CoS and QoS

Requirements for scheduling algorithms?

- An ideal scheduling discipline
  - is easy to implement
  - is fair
  - provides performance bounds
  - allows easy *admission control* decisions
    - to decide whether a new flow can be allowed
  - efficient link utilization
  - isolation between flows
  - scalability

Modified from: S. Kehav, "An Engineering Approach to Computer Networking, Addison-Wesley Professional Computing Series, 1997

Ease of implementation

- Scheduling touches every packet
- Scheduling discipline has to make a decision once every few microseconds!
- Should be implementable in a few instructions or hardware
  - for hardware: critical constraint is VLSI *space*
- Work per packet should scale less than linearly with number of active connections

From: S. Kehav, "An Engineering Approach to Computer Networking, Addison-Wesley Professional Computing Series, 1997

Ease of implementation

- However, do not fight Moore's Law
  - Decision time depends on link rate
  - Example:
    - 1500 byte packet
    - 10 Mb/s
    - Time per packet = 1.2 ms
  - Access networks have moderate speeds with moderate number of users→ complex scheduling maybe possible

Fairness

- Suppose there are n users requiring access to a link.
- The users have equal right to access the link.
- Users many have different requirements for resources
- How should resources be divided?
- Then what scheduling algorithm provides this division?

Fairness Index

- Let x_i= throughput received by i^{th} user then a fairness index is defined as

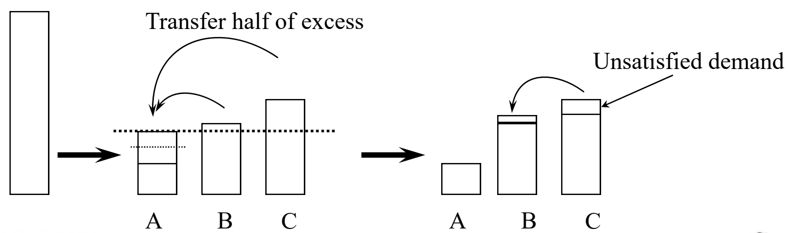
$$f(x_1, x_2 \ldots x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2} \quad 0 \le f(x_1, x_2 \ldots x_n) \le 1$$

- If $x_1 = 1, x_2 = 1 \ldots x_n = 1$ then fairness index = 1
- If $x_1 = 1, x_2 = 1, x_2 = 1 .. x_k = 1, x_{k+1} = 0 \ldots x_n = 0$

then fairness index = k/n

Fairness

- An allocation is fair if it satisfies *min-max fairness*
- Intuitively
  - each connection gets no more than what it wants
  - the excess, if any, is equally shared



THE UNIVERSITY OF KANSAS

From: S. Kehav, "An Engineering Approach to Computer Networking, Addison-Wesley Professional Computing Series, 1997

Fairness

- Formally,
  - Resources are allocated in order of increasing demand, e.g., offered load
  - No source gets a share larger than its demand
  - Sources with unsatisfied demands get an equal share of the resources

THE UNIVERSITY OF KANSAS

From: S. Kehav, "An Engineering Approach to Computer Networking, Addison-Wesley Professional Computing Series, 1997

Fairness

- Formally,

  Sources $1...n$ have resouce requirements of $x_1...x_n$.
  The link has capacity C
  Without loss in generality let $x_1 \le x_2 \le x_3 ... \le x_n$
  Give source 1 C/n ; if $C/n \ge x_1$
  then divide excess equally to the other sources.
  $C/n + (C/n - x_1)/n-1$
  If $C/n + (C/n - x_1)/n-1 \ge x_2$ repeat the process
  end where each source i gets x_i

From: S. Kehav, "An Engineering Approach to Computer Networking, Addison-Wesley Professional Computing Series, 1997

Fairness

- Example:
  - N = 4
  - x_i= 2, 2.6, 4, 5 for i=1..4
  - C =10
    - C/n = 2.5
    - 2.5 > 2 so give source 1 2units of capacity and have .5 left
    - Each now gets 2.5 + .5/4 = 2.66
    - 2.66>2.6 so give source 2 2.6 and have 0.06 left
    - 2.5 + 0.66 + 0.033=2.7 for sources 3 and 4
    - Sources 3 and 4 need 4 and 5 resources so there is no more to distribute
    - Final allocation
    - 2, 2.6, 2.7, 2.7
  - The scheduling algorithm is responsible to see that each sources gets these resources.

From: S. Kehav, "An Engineering Approach to Computer Networking, Addison-Wesley Professional Computing Series, 1997

Fairness

- What is "fair-share" if sources are not equally important
  - Each source has a weight w_i
  - Now min-max-fair share allocations is:
    - Resources are allocated in order of increasing demand, now normalized by weight
    - No source gets a share larger than its demand
    - Sources with unsatisfied demands get resources in proportion to their weights

Fairness

- Example:
  - N = 4
  - x_i= 4, 2, 10, 4 for i=1..4
  - w_i= 2.5, 4, 0.5, 1 for i=1..4
  - C =16
    - Normalize weights w_i= 5, 8, 1, 2 for i=1..4
    - View as if there are 5+8+1+2 shares to distribute, n=16 (not 4)
    - C/n = 1
      - So source 1 → 5
      - So source 2 → 8
      - So source 3 → 1
      - So source 4 → 2
    - Source 1 needs 4 so there is 1 unit of resource to distribute
    - Source 2 needs 2 so there is 6 unit of resource to distribute
    - Source 3 needs 10 so it is backlogged
    - Source 4 needs 4 so it is backlogged
    - Now have 7 units to distribute to sources 3 and 4
    - Note w_3+w_4= 3
    - Source 3 gets 7*(1/3) more units
    - Source 4 gets 7*(2/3) for 2+.7*(2/3) = 6.66 >4 that excess goes to sources 3 more units but
    - Final allocation 4, 2, 6, 4

Fairness

- Fairness is *intuitively* a good idea
- But it also tries to provides *protection*
  - traffic hogs cannot overrun others
  - automatically builds *firewalls* around heavy users
- Fairness is critical in access networks

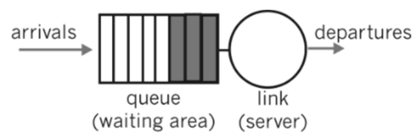
From: S. Kehav, "An Engineering Approach to Computer Networking, Addison-Wesley Professional Computing Series, 1997

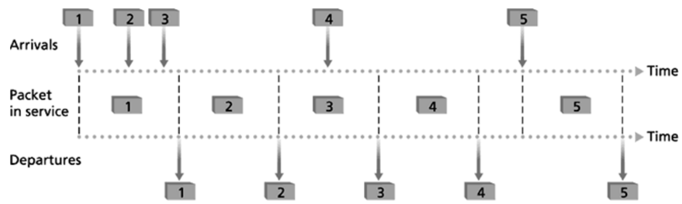
Performance bounds and Admission Control

- Performance bounds
  - Deterministic
  - Statistical
    - Probability delay > x sec is less than p
- Easy *admission control* decisions
  - Admission control needed to provide QoS
  - Overloaded resource cannot guarantee performance
  - Choice of scheduling discipline affects ease of admission control algorithm

FIFO

- Attributes
  - Simple
  - No scheduling
  - Tail dropping
  - Not *min-max fair*

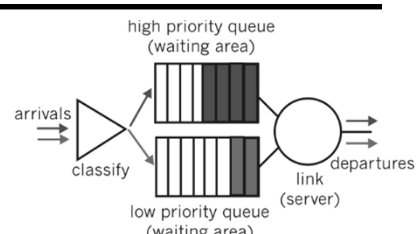


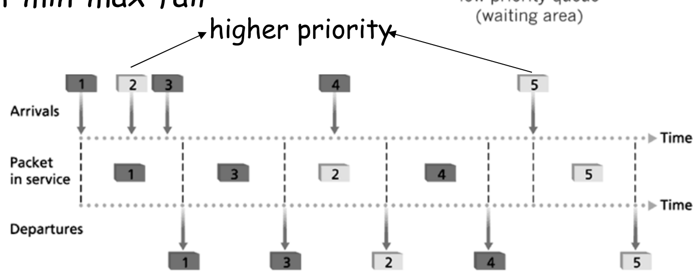


Modified from: *Computer Networking: A Top Down Approach Featuring the Internet*, 2nd edition. Jim Kurose, Keith Ross Addison-Wesley, July 2002.

Priority Queueing

- Select highest priority packet to send
- Lower priority sources can be starved out
- Not *min-max fair*





Modified from: *Computer Networking: A Top Down Approach Featuring the Internet*, 2nd edition. Jim Kurose, Keith Ross Addison-Wesley, July 2002.

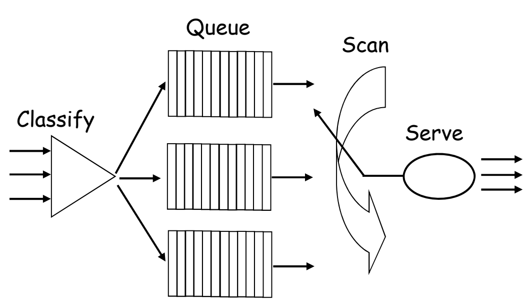
Priority Queueing

- Non-preemptive priority
  - Work-conserving
  - Complete service on packet being transmitted
- Conservation Law-
  - Delay averaged over all sources is independent of service discipline
    - Assuming work-conserving
    - Delay weighted by source load
  - So if decease delay for some sources must increase the delay for others.
- Priority only makes a difference at high loads

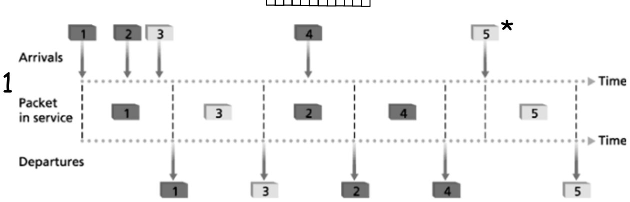
THE UNIVERSITY OF KANSAS

Round-Robin

- Cyclically scan class queues, serving one from each class
  - If queue empty then directly go to next class
- Not *min-max fair in general*
- How many packets from each queue are served in each cycle?



Packets 1, 2, 4→ Class 1
Packets 3,5→Class 2



THE UNIVERSITY OF KANSAS

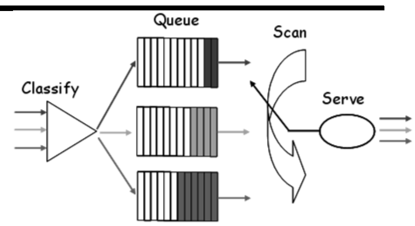
* From: *Computer Networking: A Top Down Approach Featuring the Internet*, 2nd edition. Jim Kurose, Keith Ross Addison-Wesley, July 2002.

General Processor Sharing

- *Generalized processor sharing (GPS) provides min-min fair allocation*
  - Each source has its own queue
  - Visit each non-empty queue in turn
  - Serve infinitesimal small amount of data from each queue
- GPS is unimplementable!
  - we cannot serve infinitesimals, only packets
- No packet discipline can be as fair as GPS
  - while a packet is being served, we are unfair to others
- Other scheduling disciplines attempt to approximate GPS

Weighted RR (WRR)

- If all flows
  - have same packet length
  - same weight
  - then RR is a good approximation for GSP
- If flows have different weights then serve in proportion to weight-WRR



- Example:*
  - R, G, B flows
    - have same packet lengths
    - Different weights
  - w_r= 0.5, w_G=0.75, $w_{B=}$ 1
  - Normalized weights
  - w_r= 2, w_G=3, $w_{B=}$ 4
  - Serve 2 packets from R then 3 packets from G then 4 packets from B
  - Cycle length = 9

Weighted RR (WRR)

- WRR can deal with variable sized packets by changing weights
- Example:
  - Weights→ w_r= 0.5, w_G=0.75, w_B =1
  - Average Packet Lengths→
    L_R= 50 Bytes, L_G=500 Bytes, $L_{B=}$ 1500 Bytes
  - Form modified weights →
    w_{mR} =w_R/ L_R= 0.01, w_{mG} =0.0015, w_{mB} =0.000666
  - Normalize w_r= 60, w_G=9, $w_{B=}$ 4; Cycle length = 73
  - Serve
    - 60 packets from R queue with average of 50 bytes (on average 3000 Bytes)
    - 9 packets from G queue with average of 500 bytes (on average 4500 Bytes)
    - 4 packets from B queue with average of 1500 bytes (on average 6000 Bytes)
  - Note 3000/(3000+4500+6000) =0.5/(0.5+0.75+1)
- Problems
  - Need to know average packet sizes
  - Fairness is only provided on average, i.e., over the long term
- Other scheduling disciplines address these issues.

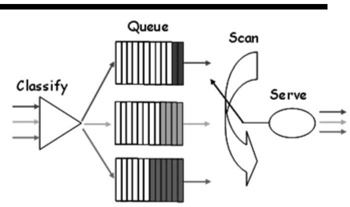
THE UNIVERSITY OF KANSAS

Weighted Fair Queueing: WFQ

- A way to view GPS is:
- Let there be N queues with weights→ $w_1 \dots w_N$
- Let NE be the set of non-empty queues at time t
- A modified weight is found as

$$w_{mi} = w_i \Big/ \left(\sum_{\forall \text{ queues in NE}} w_j \right)$$

- Serve the queues at R_i=Rw_{mi}
- So worst case, every queue has packet to send; R_i=Rw_i
- Not practical because can not serve all queues at once



- Problem: upon completion of packet transmission at time t which queue to select for next transmission?
- At time t you can determine which of the head-of-the-line packet would complete first using the GPS assuming no new arrivals
- WFQ selects this packet for transmission

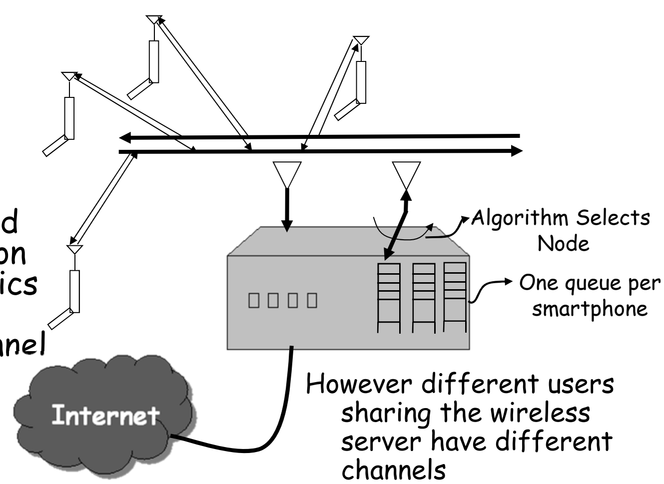
THE UNIVERSITY OF KANSAS

Weighted Fair Queueing: WFQ

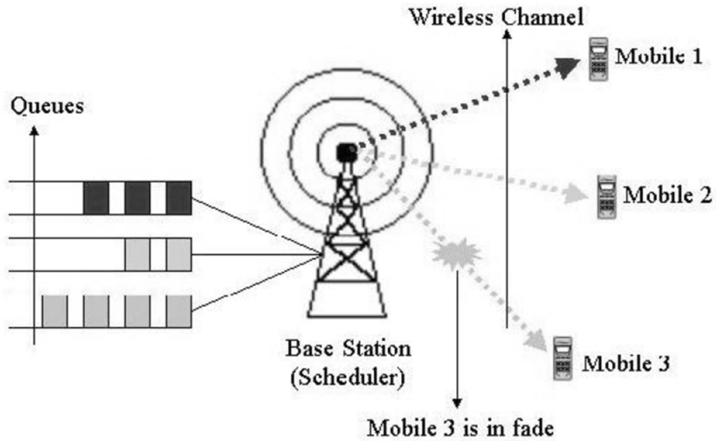
- Properties of WFQ
  - Guarantee that any packet is transmitted within packet_length/Rw_i
- Can be used to provide guaranteed services
- Achieve fair allocation
- Can be used to protect well-behaved flows against malicious flows

Interactions with Access Layer

- Pure packet scheduling techniques assume that there is no interactions between the packet selected for transmission and the dynamics of the access layer, i.e., channel conditions



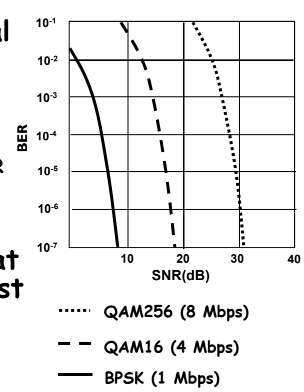
Scenario



From: Alexandre Proutiere, Ed, "QoS in multi-service wireless networks A state of the art", eurongi.enst.fr/archive/127/DWPJRA241.pdf

Wireless link characteristics

- **SNR: signal-to-noise ratio**
  - **larger SNR – easier to extract signal from noise (a "good thing")**
- **SNR versus BER tradeoffs**
  - ***given physical layer:***
  - **increase power -> increase SNR->decrease BER**
  - ***given physical layer:***
  - **decrease rate -> increase SNR->decrease BER**
  - ***given SNR:* choose physical layer that meets BER requirement, giving highest throughput**
    - **SNR may change with mobility: dynamically adapt physical layer (modulation technique, rate)**



* Modified From: *Computer Networking: A Top Down Approach Featuring the Internet*, 8nd edition. Jim Kurose, Keith Ross

Assumption

- We can measure a channel quality indicator (CQI) to estimate an achievable data rate (b/s) of a user
- A little information theory

$$\text{Channel capacity} = W \log_2(1 + \text{SNR})$$

- We can not achieve Channel Capacity
- However studies have shown that for high data rate cellular type systems achievable data rates are ~75% of the Channel Capacity*
- Thus the measured SNR can be used to determine the achievable data rate.

*T. Bonald, "Flow-level performace analysis of some opportunistic scheduling algorithms, Euro. Trans. Telecomms. 2005; 16:65–75

How do we get a CQI?

- Base station periodically send a test or pilot signal on the downlink
- Users detect pilot and use its known properties to estimate the perceived link quality, CQI
- The CQI in then fed back to the base station for use

What do we do with a CQI

- Change transmission rate to match channel conditions
  - IEEE 802.11
  - Cellular system
- Opportunistic scheduling

- Note another tool for increasing efficiency is incremental redundancy

Interactions with Access Layer

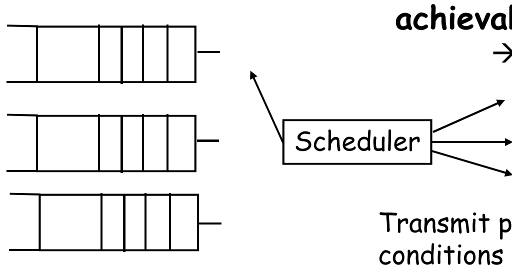
- Knowledge of the channel conditions can be factored into the scheduling algorithm to improve performance
- Resulting in "opportunistic scheduling"
- Opportunistic scheduling refers to scheduling algorithms for distributing resources in a wireless network that take advantage of instantaneous channel variations by giving priority to the users with favorable channel conditions.
- Without opportunistic scheduling maybe trying to send packets over channels that can not support the transmission; resulting is wasted resources

Interactions with Access Layer

- New model

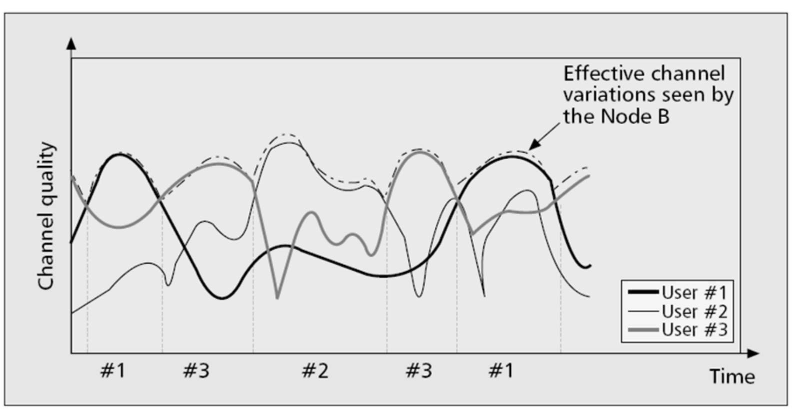
Inputs to the Scheduler:
- Queue states
- Weights
- Ability of link to support **achievable data rate R_i**

→ Link states



Transmit packets when link conditions are favorable

Interactions with Access Layer



From: Stefan Parkvall, Eva Englund, Magnus Lundevall, and Johan Torsner, "Evolving 3G Mobile Systems: Broadband and Broadcast Services in WCDMA", IEEE Communications Magazine, February 2006

Desirable properties

- Delay bound and throughput guaranteed rates.
  - Delay bound and throughput for error-free sessions are guaranteed, and are not affected by other sessions being in error.
- Long-term fairness.
  - During a large enough busy period, if a session becomes error-free, then as long as it has enough service demand, it should get back all the service “lost” while it was in error.
- Short-term fairness.
  - The difference between the normalized services received by any two error-free sessions that are continuously backlogged and are in the same state (leading, lagging, or satisfied) during a time interval should be bounded.
- Graceful degradation.
  - During any time interval while it is error-free, a leading backlogged session should be guaranteed to receive at least a minimum fraction of its service in an error-free system.

T. S. Eugene Ng, I. Stoica, H. Zhang, Packet fair queueing algorithms for wireless networks withlocation-dependent errors, In: Proc. of IEEE Infocom, 1998.

RR scheduling

- Round Robin scheduling maybe used
- RR equalizes data rates for all active users
- However, wastes radio resources

Maximum SNR (CQI) scheduler

- Assume
  - There are n active sessions 1..n
  - The base station has estimated the achievable data rate, R_i i= 1..n
- Maximum SNR scheduler selects the user j with the highest achievable data rate

  Select j where $R_j=\max\{R_1 .. R_i .. R_n\}$
- Max SNR scheduling is not fair and may starve low SNR users

KU THE UNIVERSITY OF KANSAS

Proportional Fair (PF) Scheduler

- The user with the highest achievable data rate with respect to its current mean rate gets to transmit
- PF provides a trade-off between efficiency and fairness

KU THE UNIVERSITY OF KANSAS

Proportional Fair (PF) Scheduler

- Each user k, at time slot t
- Current Rate $R_k[t]$ (from data rate control message-DRC or CQI)
- Time-averaged Rate $A_k[t]$

$$A_k[t+1] = \begin{cases} (1-\alpha)A_k[t] + \alpha R_k[t] & \text{if k is scheduled in slot t+1} \\ (1-\alpha)A_k[t] & \text{if k is not scheduled in slot t+1} \end{cases}$$

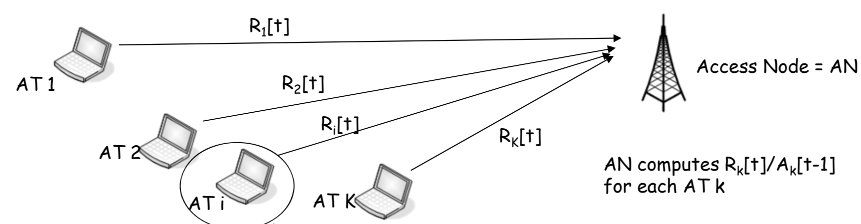
α determines the time constant of the algorithm

- User with maximum $R_j[t]/A_j[t]$ is scheduled

$$\text{Select j}^{\text{th}} \text{ user where } \frac{R_j[t]}{A_j[t]} = \max\left\{\frac{R_1[t]}{A_1[t]} \cdots \frac{R_i[t]}{A_i[t]} \cdots \frac{R_n[t]}{A_n[t]}\right\}$$

KU THE UNIVERSITY OF KANSAS

How PF scheduler works



Each AT k reports DRC $R_k[t]$ at time slot t

Slot allocated to AT i

AN computes $R_k[t]/A_k[t-1]$ for each AT k

AN allocates time slot to AT with maximum $R_k[t]/A_k[t-1]$, say AT i

Exponential weighted average throughput to each AT

$$A_k[t] = \begin{cases} (1-\alpha)A_k[t-1] + \alpha R_k[t] & \text{if k is scheduled in slot t} \\ (1-\alpha)A_k[t-1] & \text{if k is not scheduled in slot t} \end{cases}$$

$A_i[t]$ of AT i updates as

$$A_i[t] = (1-\alpha)A_i[t-1] + \alpha R_i[t]$$

$A_k[t]$ of all other ATs k updated

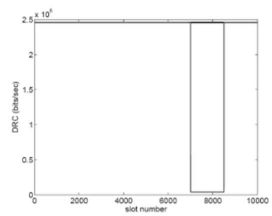
$$A_k[t] = (1-\alpha)A_k[t-1]$$

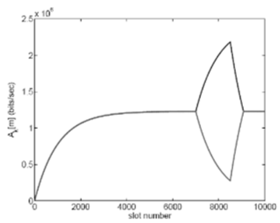
Schedule AT that has its better than average conditions i.e., schedule AT with maximum R/A

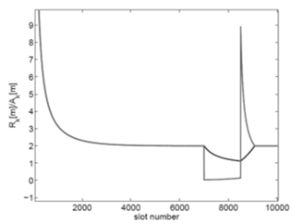
KU THE UNIVERSITY OF KANSAS

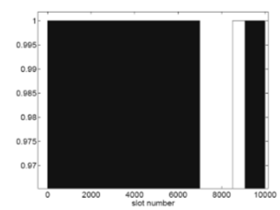
PF Scheduler: Example

- Both AT1 and AT2 have R of of 2.4Mbps
- Each AT gets half the slots (1.2Mbps)
- AT2 experiences fading
- AT1 gets all the slots when AT2 is in fading
- This improves sector throughput





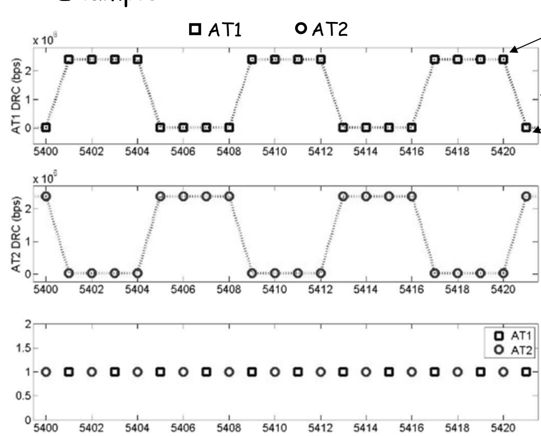




THE UNIVERSITY OF KANSAS

Why not round-robin scheduling?

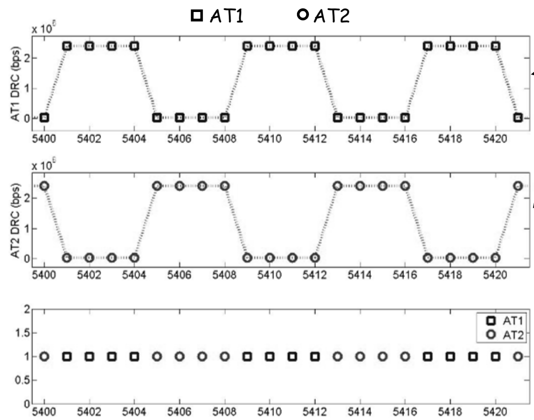
- Ensures fairness but can be sub-optimal (not channel aware)
- Example: -



AT1 throughput = 600Kbps
AT2 throughput = 600Kbps
System throughput = 1.2Mbps
Fairness: ½ time slots

Proportional fair (PF) scheduler

- PF is channel aware
- Improves system throughput while maintaining fairness(1)
- Schedule AT that is experiencing better than avg. conditions



AT1 DRC
AT2 DRC

AT1 throughput = 1.2Mbps
AT2 throughput = 1.2Mbps
System throughput = 2.4Mbps
Fairness: ½ time slots

System throughput higher because all slots can be used

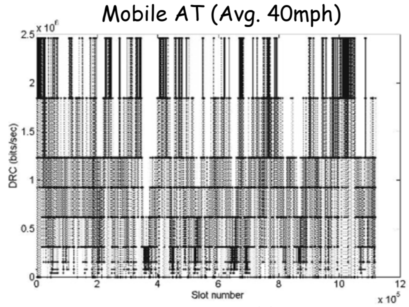
Decision: which AT got the time-slot

(1) Under i.i.d. channel conditions

Data rate traces

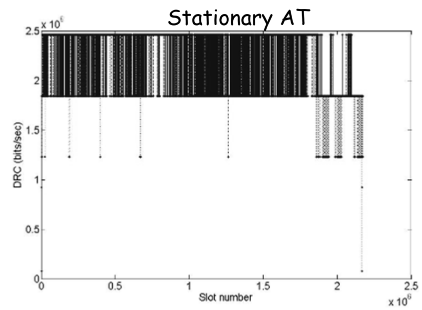
- When DRC variance high, more benefit in using PF (compared to round-robin)
- Two DRC traces collected using Qualcomm CDMA Air Interface Tester (CAIT) – DRC reported in each time-slot by the laptop
- Mobile: driving from Burlingame to Palo Alto (Avg. 40mph)
- Stationary: laptop on desk at Burlingame

Mobile AT (Avg. 40mph)



~30 minutes

Stationary AT



~60 minutes

PF Scheduler Properties

- Improves sector throughput
  - Schedules ATs in their better than average channel conditions
- If channel conditions IID
  - Long-term fairness achieved
    - Assume infinite backlog
  - Maximizes sum(log(A_k))

KU THE UNIVERSITY OF KANSAS

Packet scheduling: General Discussion

- Let:
  - $R_k[t]$ = instantaneous data rate obtained from feedback for k^{th} user = R_k
  - $A_k[t]$ = average throughput for k^{th} user = A_k
  - For convenience drop t
  - $U_k(R_k)$ = Utility function
  - U=sum ($U_k(R_k)$)
  - Want to max(U)

KU THE UNIVERSITY OF KANSAS

Packet scheduling: General Discussion

$$A_k[t] = \begin{cases} (1-\alpha)A_k[t-1] + \alpha R_k[t] & \text{if k is scheduled in slot t} \\ (1-\alpha)A_k[t-1] & \text{if k is not scheduled in slot t} \end{cases}$$

- M_k=scheduling metric = $R_k dU_k/dA_k$
- Now
  - For RR
    - $U_k(A_k) = 1$
    - $M_k=0$
  - For PFQ
    - $U_k(A_k) = \log(A_k)$
    - $M_k=R_k/A_k$

KU THE UNIVERSITY OF KANSAS

Packet scheduling: General Discussion

- For Maximum SNR scheduler
  - $U_k(A_k) = A_k$
  - $M_k = R_k$
- Minimum guaranteed bit rate scheduling (min-GBR) controls how much preference is given to the users where their bit rates drops below guarantee bit rate GBR
  - $U_k(A_k) = A_k + (1-\exp(-\beta(A_k- A_{min}))$
  - $M_k = R_k(1-\exp(-\beta(A_k- A_{min})))$
  - A_{min}= GBR (target minimum bit rate)

KU THE UNIVERSITY OF KANSAS

Packet scheduling: General Discussion

- Minimum guaranteed bit rate scheduling (min-GBR) with PFQ controls how much preference is given to the users where their bit rates drops below GBR with PFQ
  - $U_k(A_k) = \log(A_k) + (1-\exp(-\beta(A_k - A_{min}))$
  - $M_k = R_k(1/A_k - \beta\exp(-\beta(A_k - A_{min})))$
  - A_{min}= GBR (target minimum bit rate)

Packet scheduling: General Discussion

- Control the delay of the head of line packet based on a maximum delay specification.
  - $U_k(A_k) = -\log(\delta_n) \log(A_k) d_{HOL,n} / d_{Req,n}$
  - $M_k = R_k(-\log(\delta_n) d_{HOL,n} / A_k d_{Req,n}$
  - $d_{HOL,n}$= Head of Line packet delay
  - $d_{Req,n}$ = Maximum delay specification
  - δ_n = Aggressiveness factor

Packet scheduling: General Discussion

- How do you deal with delay and packets that are waiting to retransmit?
- Alternatives
  - Across all users (k):
    - Consider all users with pending retransmissions. Send these with priority; if multiple users have pending transmissions then use one of the above to select next packet to transmit
    - Better from a delay perspective
  - Within each flow:
    - use one of the above to select next user to get a chance to transmit, send pending retransmissions first before new transmissions
    - Better from a capacity perspective
- In practice not much performance difference

THE UNIVERSITY OF KANSAS

Other Schedulers

- Optimum Channel-Aware Scheduling with Differentiation (OCASD)
  - Optimizes trade-off between
    - Short term fairness
    - Delay
    - Maximum throughput
- Best Link Lowest Throughput First (BLOT)
  - Optimizes trade-off between
    - Throughput and fairness
    - Guarantees minimum service
    - Maintains stability
- Others.....

THE UNIVERSITY OF KANSAS