King Fahd University of Petroleum & Minerals Computer Engineering Dept

COE 402 – Computer System Performance – Project Statement

Term 043

Dr. Ashraf S. Hasan Mahmoud

Rm 22-148-3

Ext. 1724

Email: ashraf@ccse.kfupm.edu.sa

Lecture Contents

1.

CDMA Downlink Capacity – Problem Statement

- This is a data transmission scheduling problem where the total system power is the shared resource. CDMA is the chosen multiple access for future mobile and wireless networks. For CDMA systems there exist no hard capacity limit (i.e. number of time slots or frequency carriers) but rather connection requests are accepted as long as the collective quality of all transmissions within the network is acceptable. For downlink (from basestation to mobile) transmissions, the total available power for traffic should not be exceeded as well. Under these conditions and using basic scheduling techniques like FCFS, it is required to evaluation the network throughput and other basic performance figures like mean access delay, request drop rate, etc.

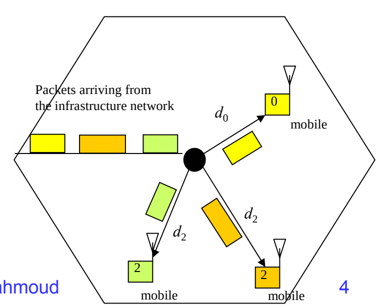
System Description

- System:
  - Basestation with maximum power budget of Ptotal Watts
  - Can support only the following rates: R0, R1 = 2R0, R2 = 4R0, R3 = 8R0, and R4 = 16R0
  - System bandwidth = W Hz
  - Orthogonality factor = ρ (0 ≤ ρ ≤ 1)

Figure shows the cell of interest with a basestation in the center. The basestation is communicating on the downlink with three mobiles located at distances: d_0, d_1, and d_2 from the basestation.

The basestation receives from the infrastructure network packets that are destined for each of users. It transmits these packets on the downlink using the assigned system rates.

When the base station can not serve an incoming packet, the packet is queued at basestation till it is possible to transmit the packet.



System Description (2)

- Consider a basestation in the cell of interest, serving N users
- i^{th} user (i=0, 1, …, N) is at distance d_i from basestation
- The i^{th} user is assigned a data rate R_i

- The simultaneous downlink transmissions from one basestation are ideally orthogonal (i.e. do not interfere with each other). However, in practice, there is loss of orthogonality and therefore, simultaneous downlink transmissions interfere with one another
  - ρ = 1 means perfect orthogonality
  - ρ = 0 means total loss of orthogonality

System Description (3)

- The basestation can assign any of the follow bit rates ={R_i, $2^i R_0$, i=0, 1, …, 5} where R_0 = 9.6 kb/s –
- Assume users request the highest rate possible given the available power and interference conditions
- ➔ The basestation allocates enough transmit power, P_i, to the i^{th} mobile such that recevied (Eb/No)i is at least equal to a required EbNo threshold.

System Description (4)

- The basestation is continuously transmitting overhead channels (pilot, sync and paging) – The fraction of power allocated for overhead channels is β (0≤β≤1)
  - Overhead power, Pov = β Ptotal
  - The maximum power that can be used for traffic channels is given by, Ptraffic = (1-β)Ptotal
- Thermal noise can be neglected

System Description (5)

- When our cell is serving N transmissions (or bursts), then the total transmitted power is given by:

$$\sum_{i=0}^{N-1} P_i + Pov$$

which is always less than or equal to Ptotal

What Are The Typical System Parameters?

- System bandwidth, W = 1.25 MHz
- Minimum Eb/No required = 5 dB (note has to be converted to real numbers – i.e. 10^(5/10) = 3.1623)
- Overhead power fraction, β = 0.2
- Total power budget = 24 Watts
- Orthogonality factor, ρ = 0.1

- Mean packet arrival per user = 5 packets/sec
- (mean) packet size = 1000 bit

How to Allocate P_i?

- For the ith user (i=0,1, …, N-1) calculate g_i, where g_i is given by
  $$g_i = \left(\left(W/R_i \right) / \left(E_b/N_0 \right)_{\min} + 1 - \rho \right)^{-1}$$
  - Note (Eb/No)min is the minimum required Eb/No figure for correct operation of the link
  - ρ is the orthogonality factor
  - W is the system bandwidth in Hz
- Perform the following check $\sum_{i=0}^{N-1} g_i \left(1 - \rho + f_i\right) \le \left(1 - \beta\right)$
  - Remember β is the fraction of overhead power
- If the above is true then Pi's can be calculated using
  $$P_i = g_i \left(1 - \rho + f_i\right) \frac{Pov}{1 - \sum_{i=0}^{N-1} g_i \left(1 - \rho + f_i\right)}$$
  - f_i is defined in next slide
- If the condition is NOT true, then the system can not support these specific N users at the specified rates

Definition of f_i

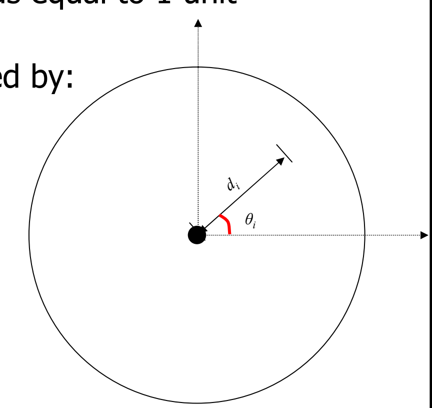
- f_i is a parameter that summarizes the radio frequency propagation loss and intercell interference
  - This parameter is a function of the user location (distance)
- For a user at distance d_i from basestation, f_i for that particular user is equal to

$$f_i(d_i) = \sum_{k=0}^{5} a_k d_i^k$$

where the coefficients of the 5[th] degree polynomial are given by
$[a_5\ a_4\ a_3\ a_2\ a_1\ a_0]$ = [17.04825264741491
-28.61622283071572
20.16491247631064
-5.67993466014081
0.61065458307801
0];

How to Position Users?

- To position user's randomly (and uniformly) in the cell
- Assume a circular cell of radius equal to 1 unit
- The position (d_i, θ_i) is specified by:
  - d_i = sqrt(rand())
  - θ_i = 2*π * rand()



Only the distance d_i is important for our calculations. Given d_i, you can
- Calculate f_i (a 5[th] degree polynomial in d_i),
- Using f_i , the required (Eb/No)min, and the bit rate R_i to be assigned, g_i (and P_i if possible) can be calculated.

How to Generate Packet Arrivals for One User?

- Remember all packet arrivals happen at the basestation.
- Basestation attempts to assign a bit rate for the user and transmit the packet to the user on the downlink
- Interarrival times for packets of one user can be generated by:
  - Interarrival_Time = - MeanInterArrivalTime * log(rand());
  - MeanInterArrivalTime = 1/MeanArrivalRate

- E.g if MeanArrivalRate = 5 packets/sec ➔ MeanInterarrivalTime = 0.2 sec/packet
  - However, specific interarrival times are randomly generated
  - For example, 1st 10 samples may be: 0.1412 0.1771 0.0040 0.1207 0.0777 0.1350 0.0374 0.0405 0.3027 0.4783
  - The mean for these 10 samples is 0.1514 which is not exactly 0.2
  - If we generate 1000 samples, the mean may be 0.1970 which now closer to 0.2
- Usually, the problem specifies the mean packet arrival rate and not the mean interarrival time.

What about the Size of Each Packet Arrival?

- We can also make the size random (in a manner similar to that we used for the interarrivals)

- However, for the time being – use fixed 1000 bit/packet sizes.

Performance Metrics

- Required output:
  - Throughput – time average of assigned rates (see next slide)
  - Probability of burst request queueing – Ratio of queued burst requests to total number of number of arriving requests
  - Average access delay for a burst request
  - When a time-out timer is implemented, Probability of burst request dropping can be evaluated – Ratio of dropped burst requests to total number of number of arriving requests
  - Etc.

Needed Data Structures

- Three main structures are needed
- (1) Channel structure – holds the up to date view of currently active transmissions in the system
- (2) Arrivals structure – holds the next arrival for every user in the system
- (3) Queue structure – hold the requests that could not be served

Simultaneous Channel Connections				
Channel Id	1	2	3	...
User Id				
Packet Id				
Rate				
Power				
PacketSize				
Start Time				
Finish Time				
Total Power				
Total Rate				

Queued Requests		
Index	User Id	Arrival Time
1		
2		
3		
.		
.		

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1		
2		
..		
N		

Manual Simulation - 1

- Initially, the channel and queue are empty
- Allocate N users (as shown in slide 12) – calculate their corresponding di and fi
  - Assume N = 4
  - di = [0.9747 0.4808 0.7790 0.6971] (di = sqrt(rand());)
  - fi = [3.0416 0.1306 0.9140 0.5453]
- Calculate the first arrival for every user (Interarrival_Time = -MeanInterArrivalTime * log(rand());)
- What is the earliest event in the system?
  - User 3, Packet 1 arriving at t = 0.2
  - System Time ← 0.2
  - Note – no channel or queued events

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	0.301	1
2	1.2	1
3	0.2	1
4	0.7	1

Manual Simulation - 2

- User 3 (Packet 1) – d3 = 0.7790, f3 = 0.9140
- What rate can be assigned
  - Rate 153.6 kb/s can be assigned – finish time = System Time + 1000/153.6kb/s = 0.2065 sec
- Update channel structure
- Get next arrival for user 3

This is a temporary structure used to determine the rate assignment and overall power assignment once the new packet is served (if possible)

Rate Assignment	
Channel Id	1
User Id	3
Rate	307.2
di	0.7790
fi	0.9140
gi	0.4573
Σgi(1-ρ+fi)	0.830 > 0.8
Pi	Can not

Trying rate 307.2 kb/s-

Rate Assignment	
Channel Id	1
User Id	3
Rate	153.6
di	0.7790
fi	0.9140
gi	0.2879
Σgi(1-ρ+fi)	0.5222 ≤ 0.8
Pi	5.2469

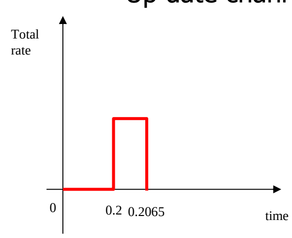
Trying rate 153.6 kb/s-

Simultaneous Channel Connections	
Channel Id	1
User Id	3
Packet Id	1
Rate	153.6
Power	5.2469
PacketSize	1000
Start Time	0.2
Finish Time	0.2065
Total Power	5.2469
Total Rate	153.6

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	0.301	1
2	1.2	1
3	0.3	2
4	0.7	1

Manual Simulation - 3

- What is the next event?
  - Earliest arrival is from user 3 (at t = 0.3 sec)
  - Channel release (at t = 0.2065 sec)
  - ➔ next event is channel release – System Time = t = 0.2065
- Channel release event:
  - Release (free) power
  - Up date channel structure

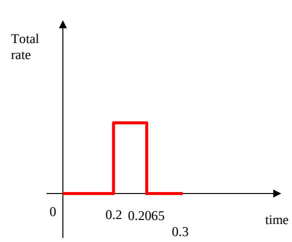


Simultaneous Channel Connections	
Channel Id	Empty
User Id	
Packet Id	
Rate	
Power	
PacketSize	
Start Time	
Finish Time	
Total Power	
Total Rate	

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	0.301	1
2	1.2	1
3	0.3	2
4	0.7	1

Manual Simulation - 4

- What is the next event?
  - Earliest arrival is from user 3 (at t = 0.3 sec)
  - No channel events
  - Next event is arrival from user 3 – System Time = 0.3



Simultaneous Channel Connections	
Channel Id	Empty
User Id	
Packet Id	
Rate	
Power	
PacketSize	
Start Time	
Finish Time	
Total Power	
Total Rate	

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	0.301	1
2	1.2	1
3	0.3	2
4	0.7	1

Manual Simulation - 5

- User 3 (Packet 2) – d3 = 0.7790, f3 = 0.9140
- What rate can be assigned
  - Rate 153.6 kb/s can be assigned – finish time = System Time + 1000/153.6kb/s = 0.3065 sec
- Update channel structure
- Get next arrival for user 3

This is a temporary structure used to determine the rate assignment and overall power assignment once the new packet is served (if possible)

Rate Assignment	
Channel Id	1
User Id	3
Rate	307.2
di	0.7790
fi	0.9140
gi	0.4573
Σgi(1-ρ+fi)	0.830 > 0.8
Pi	Can not

Trying rate 307.2 kb/s-

Rate Assignment	
Channel Id	1
User Id	3
Rate	153.6
di	0.7790
fi	0.9140
gi	0.2879
Σgi(1-ρ+fi)	0.5222 ≤ 0.8
Pi	5.2469

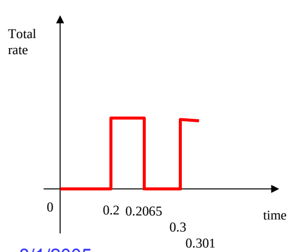
Trying rate 153.6 kb/s-

Simultaneous Channel Connections	
Channel Id	1
User Id	3
Packet Id	1
Rate	153.6
Power	5.2469
PacketSize	1000
Start Time	0.3
Finish Time	0.3065
Total Power	5.2469
Total Rate	153.6

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	0.301	1
2	1.2	1
3	0.5	3
4	0.7	1

Manual Simulation - 6

- What is the next event?
  - Next arrival User 1 (packet 1) at t = 0.301
  - Channel release at t = 0.3065
  - ➔ next event is arrival from user 1 – System Time = 0.301



Simultaneous Channel Connections	
Channel Id	1
User Id	3
Packet Id	1
Rate	153.6
Power	5.2469
PacketSize	1000
Start Time	0.3
Finish Time	0.3065
Total Power	5.2469
Total Rate	153.6

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	0.301	1
2	1.2	1
3	0.5	3
4	0.7	1

Manual Simulation - 7

- User 1 (Packet 1) – d3 = 0.9747, f3 = 3.0416
- What rate can be assigned to user 1?
  - Remember User 3 is already on the air and it is assigned 153.6 kb/s
  - 19.2 kb/s can be assigned – finish time = System Time + 1000/19.2kb/s = 0.3531 sec

Rate Assignment		
Channel Id	1	2
User Id	3	1
Rate	153.6	307.2
di	0.7790	0.9747
fi	0.9140	3.0416
gi	0.2879	0.4573
Σgi(1-ρ+fi)	2.3247 > 0.8	
Pi	N.A.	N.A.

Trying rate 307.2 kb/s-

Rate Assignment		
Channel Id	1	2
User Id	3	1
Rate	153.6	153.6
di	0.7790	0.9747
fi	0.9140	3.0416
gi	0.2879	0.2879
Σgi(1-ρ+fi)	1.6570 > 0.8	
Pi	N.A.	N.A.

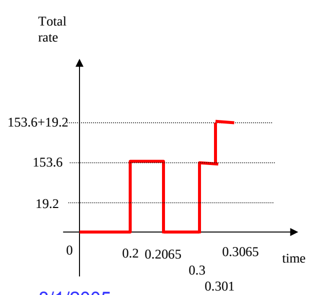
Trying rate 153.6 kb/s-

Rate Assignment		
Channel Id	1	2
User Id	3	1
Rate	153.6	19.2
di	0.7790	0.9747
fi	0.9140	3.0416
gi	0.2879	0.0465
Σgi(1-ρ+fi)	0.7057 ≤ 0.8	
Pi	8.5169	2.9915

Trying rate 19.2 kb/s-

Manual Simulation - 8

- Update channel structure
- Get next arrival for user 1



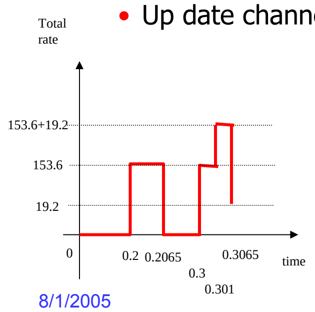
Simultaneous Channel Connections		
Channel Id	1	2
User Id	3	1
Packet Id	1	1
Rate	153.6	19.2
Power	8.5169	2.9915
PacketSize	1000	1000
Start Time	0.3	0.301
Finish Time	0.3065	0.3531
Total Power	11.5084	
Total Rate	172.8	

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	0.34	2
2	1.2	1
3	0.5	3
4	0.7	1

Note: when user 3 ended its transmission, the powers assigned to other ongoing transmissions are updated (decrease!!)

Manual Simulation - 9

- What is the next event?
  - Earliest arrival User 1 at t = 0.34
  - Earliest channel release time t = 0.3065
  - ➔ channel release event – System Time = 0.3065
- Channel release event:
  - Release (free) power
  - Up date channel structure



Simultaneous Channel Connections		
Channel Id	1	2
User Id	Empty	1
Packet Id		1
Rate		19.2
Power		1.0783
PacketSize		1000
Start Time		0.301
Finish Time		0.3531
Total Power	1.0783	
Total Rate	19.2	

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	0.34	2
2	1.2	1
3	0.5	3
4	0.7	1

Note: No change to assigned rate or power!!

Manual Simulation - 10

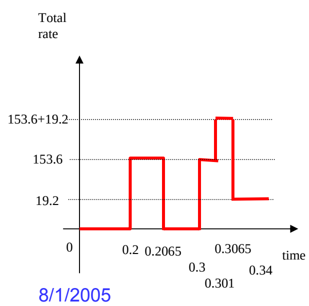
- What is the next event?
  - Earliest arrival User 1 at t = 0.34
  - Earliest channel release time t = 0.3531
  - ➔ Arrival event – System Time = 0.341
- Process Arrival Event
  - User already assigned a rate – update finish time
  - Get Next arrival for User 1

Remaining bits = PacketSize - (SystemTime – StartTime)XRate
= 1000 – (0.34 – 0.301)X19200 = 251.2 bit
NewPacketSize = NewArrival + RemainingBits
= 1000 + 251.2 = 1251.2 bit
New FinishTime = SystemTime + NewPacketSize/Rate
= 0.34 + 1251.2/19200 = 0.4052

Simultaneous Channel Connections		
Channel Id	1	2
User Id	Empty	1
Packet Id		1
Rate		19.2
Power		1.0783
PacketSize		1251.2
Start Time		0.34
Finish Time		0.4052
Total Power	1.0783	
Total Rate	19.2	

Manual Simulation - 11

- The channel and arrivals structure are updated



Simultaneous Channel Connections		
Channel Id	1	2
User Id	Empty	1
Packet Id		1
Rate		19.2
Power		1.0783
PacketSize		1251.2
Start Time		0.34
Finish Time		0.4052
Total Power	1.0783	
Total Rate	19.2	

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	2.0	3
2	1.2	1
3	0.5	3
4	0.7	1

Manual Simulation - …

- …

Manual Simulation - … +1

- Assume the current system state is as depicted by the main data structures
- What is the next event?
  - Earliest arrival User 1 (Packet 45) t = 5.311
  - Earliest channel release t = 5.3133
  - ➔ Next event is arrival from User 1 – System Time = 5.311

Simultaneous Channel Connections		
Channel Id	1	2
User Id	4	3
Packet Id	53	30
Rate	307.2	19.2
Power	12.4595	1.5914
PacketSize	1000	1000
Start Time	5.31	5.30
Finish Time	5.3133	5.3521
Total Power	14.0509	
Total Rate	326.4	

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	5.311	45
2	5.312	28
3	6.2	31
4	7.1	54

Manual Simulation - … +2

- User 1 (Packet 1) – d3 = 0.9747, f3 = 3.0416
- What rate can be assigned to user 1?
  - Remember Users 4 and 3 is already on the air
  - User can not be assigned any rate ➔ Queue packet arrival
- Update queue structure

Rate Assignment			
Channel Id	1	2	3
User Id	4	3	1
Rate	307.2	19.2	307.2
di	0.7790	0.9747	0.9747
fi	0.5453	0.9140	3.0416
gi	0.4573	0.0465	0.4573
Σgi(1-ρ+fi)	2.5479 > 0.8		
Pi	N.A.	N.A.	N.A

Trying rate 307.2 kb/s-

Rate Assignment			
Channel Id	1	2	3
User Id	4	3	1
Rate	307.2	19.2	9.6
di	0.7790	0.9747	0.9747
fi	0.5453	0.9140	3.0416
gi	0.4573	0.0465	0.0238
Σgi(1-ρ+fi)	0.8390 > 0.8		
Pi	N.A.	N.A.	N.A

Trying rate 9.6 kb/s-

User can not be served even at the smallest system rate – queue packet arrival

Manual Simulation - … +3

- Updated channel, arrivals, and queue structures

Queued Requests		
Index	User Id	Arrival Time
1	1	5.311

Simultaneous Channel Connections		
Channel Id	1	2
User Id	4	3
Packet Id	53	30
Rate	307.2	19.2
Power	12.4595	1.5914
PacketSize	1000	1000
Start Time	5.31	5.30
Finish Time	5.3133	5.3521
Total Power	14.0509	
Total Rate	326.4	

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	8.4	46
2	5.312	28
3	6.2	31
4	7.1	54

An arriving request is queued if queue is not empty!!

Manual Simulation - … +4

- What is the next arrival?
  - Earliest arrival User 2 (packet 28) – t = 5.312
  - Earliest channel release – t = 5.3133
  - ➔ Next event is an arrival from user 2 – System Time = 5.312
- Process Arrival event:
  - Since queue is not empty and we are assuming a strict FCFS
  - Queue arrival
- Update structures

Queued Requests		
Index	User Id	Arrival Time
1	1	5.311
2	2	5.312

Simultaneous Channel Connections		
Channel Id	1	2
User Id	4	3
Packet Id	53	30
Rate	307.2	19.2
Power	12.4595	1.5914
PacketSize	1000	1000
Start Time	5.31	5.30
Finish Time	5.3133	5.3521
Total Power	14.0509	
Total Rate	326.4	

Next Packet Arrivals		
User Id	Arrival Time	Packet Id
1	8.4	46
2	7.3	29
3	6.2	31
4	7.1	54

After a channel release event, queued requests are served till queue is empty or blockage

Manual Simulation - … +5

- What is the next arrival?
  - Earliest arrival User 3 (packet 31) – t = 6.2
  - Earliest channel release – t = 5.3133
  - ➔ Next event is channel release – System Time = 5.3133
- Process channel release:
  - Since queue is not empty – then system should attempt to serve queued requests until queue is empty or blocked
- Update structures

Simultaneous Channel Connections

Channel Id	1	2	3
User Id	1	3	2
Packet Id	45	30	28
Rate	76.8	19.2	19.2
Power	1.8778	14.4992	1.0669
PacketSize	1000	1000	1000
Start Time	5.3133	5.30	5.3133
Finish Time	5.3263	5.3521	5.3654
Total Power	17.4439		
Total Rate	115.2		

Next Packet Arrivals

User Id	Arrival Time	Packet Id
1	8.4	46
2	7.3	29
3	6.2	31
4	7.1	54

Queued Requests

Index	User Id	Arrival Time
1	Empty	
2		

Manual Simulation - …

- Etc.

How to Determine Next Event?

- T1 = earliest arrival event
- T2 = earliest channel release time

```
If (T1 < T2)
  Then NextEvent   = ArrivalEvent
       SystemTime = T1
  else NextEvent   = ChannelRelease
       SystemTime = T2
  end
```

- System Time is ALWAYS earliest arrival event or earliest channel release

How to Process Arrival Events?

```
if (User already on air)
then compute remaining bits from previous transmission
     update start time, packet size, and finish time
else if (Queue is Empty)
     then determine maximum rate to assign
     if (no rate can be assigned)
     then queue arrival
     else assign computed rate
          calculate finish time
     end
     else queue arrival
end
```

How to Process Channel Release Events?

```
• Free corresponding channel structure
  if (Queue is NOT Empty)
  then while (Queue is NOT Empty)
         determine rate to serve queued request
         if (no rate can be assigned)
         then break
         else assign rate and calculate finish time
         free queued element
         get next element
       endwhile
  end
  Update channel structure
```


Main Routine

```
// This is the main routine in the simulation code
Main(){
   InputNoOfTerminals( );
   for i=1 to NoOfIterations
      InitializeDataStructures( );
      ClearIterationStatistics( );
      SimulateSystem( NoOfTermainls );
      CollectIterationStatistics( );
   end
   AverageAllIterationsStatistics( );
   OutputStatistics( )
}
```

InitializeDataStructures Routine

```
InitializeDataStructures()
{
 // In this routine the three main
   data structures: Arrivals,
   ChannelStatus, and Queue are
   created and initialized properly
 // The three structures are shown in
   slide 16
}
```


ClearIterationStatistics Routine

```
ClearIterationStatistics()
{
 // In this routine the variables used to accumulate
   quantities of interest like total throughput and
   total request delay, etc. are cleared – These
   will be used in the CollectIterationStatistics

  AccumulateTotalThroughput = 0;
  AccumulateTotalDelay      = 0;
  NoOfServedRequests        = 0;
  NoOfQueuedRequests        = 0;
  .
  .
  .
}
```

SimulateSystemRoutine

```
SimulateSystem( int NoOfTermainls )
{
 // This is the main simulation routine where a number of terminals are
    located in the cell at random distances – and then the simulation is
    conducted till MaxSimulationTime is reached

    LocateTerminalsInCell( NoOfTermainls );
    CalculatefiForTerminals( NoOfTermainls );
    CreateFirstArrivals( NoOfTermainls );
    SystemTime = 0.0;
    while (SystemTime <= MaxSimulationTime){
        // This is routine gets the next event to process and assigns its
        // value to the SystemTime variable – see slide 35 for details on
        // how to determine SystemTime
        // Remember, we have two events only – Request arrival event and
        // Channel release event
        GetNextEvent( &SystemTime, &EventType );
         if (EventType == ARRIVAL_EVENT)
           ProcessArrivalEvent( );        // flow chart in slide 36
        else
           ProcessChannelReleaseEvent( ); // flow chart in slide 37
    }
}
```


Required Metrics and How to Calculate Them

- Average system throughput = average area under the total sum of rates versus time curve
- Average system power used = average area under the total power used versus time curve
- Average queue length = average area under the queue size versus time curve
- Probability of being delayed = No of packets that are queued relative to number of total packets generated
- Average access delay = mean of packet waiting time for all packets

How Did I do the previous Calculations?

```
clear all
% initialize system paramters
W    = 1.25e6; % bandwidth is 1.25 MHz
EbNo_dB = 5;   % minimum required EbNo in dBs
EbNo = 10^(EbNo_dB/10); % converting to real numbers
Rho  = 0.1;    % Orthogonolity factor
Beta = 0.2;    % fraction of overhead power
Ptotal = 24;   % total power at basestation
Pov    = Beta*Ptotal; % power for overhead channels
SystemRates = [9.6  19.2  38.4  76.8  153.6  307.2]*1e3; % all possible system rates
d = [0.9747    0.4808    0.7790    0.6971]; % Four users located (distances) randomly
f = f_function_circle_05_S6(d);             % their corresponding fi

% The routine below takes rates in vector "Rates" and the corresponding fi
% for the users in the vector "f_i" and performs the power assignment
% calculations
% Example1:
% To check whether we can assign user 1 and 3 the bit rates
% 9.6kb/s and 153.6kb/s, the "Rates" and "f_i" vector should look like:
% Rates = [SystemRate(1) SystemRate(5)], and f_i = [f(1) f(3)]
%
% Example2:
% To check whether we can assign user 4 and 2 the bit rates
% 307.2kb/s and 78.8kb/s, the "Rates" and "f_i" vector should look like:
% Rates = [SystemRate(6) SystemRate(4)], and f_i = [f(4) f(2)]
%
% Example3:
% To check whether we can assign user 1 the bit rate
% 307.2kb/s, the "Rates" and "f_i" vector should look like:
% Rates = [SystemRate(6)], and f_i = [f(1)]
%
```

Use this code for the power assignments

How Did I do the previous Calculations? (2)

```
Rates = [SystemRates(4)  SystemRates(5)];
f_i   = [f(1)    f(3)];
gi = 1./((W./Rates)/EbNo + 1 - Rho);  % calculate gi's
Gi = gi.*(1-Rho+f_i); % Calculate Gi = gi*(1-Rho+fi)
SumGis = sum(Gi); % sum all Gi's to test feasiblity
fprintf('SumGis = %7.4f\n', SumGis);
if (SumGis <= (1-Beta))
    fprintf('Assigment can be performed\n');
    Pi = gi.*(1-Rho+f_i) .* Pov./(1-SumGis);
    fprintf('gi = %7.4f \n', gi);
    fprintf('power assignment = %7.4f\n', Pi);
    %
    % as a test - all computed EbNos after the power assignment should be
    % equal to the minimum required EbNo specified above
    EbNo_t = W./Rates .* (Pi./((1-Rho+f_i).*(sum(Pi)+Pov) - (1-Rho)*Pi));
else
    fprintf('Assignment can not be performed - SumGis is GREATER THAN (1-
  Beta)\n');
end
```

BACKGROUND MATERIAL (NOT REQUIRED)

Fixed Assignment Access for Voice-Oriented Networks

- Voice-Oriented networks ~ cellular telephony or PCS
- Fixed allocation of resource (frequency, time, code, etc)
- Three basic access techniques:
  - Frequency Division Multiple Access (FDMA)
  - Time Division Multiple Access (TDMA)
  - Code Division Multiple Access (CDMA)
- The choice of technology impacts:
  - Capacity
  - QoS

Background and References

- Background:
  - Chapter 8, W. Stallings, "Data and Computer Communications," Sixth Edition, Prentice Hall International Inc.
- References:
  - Chapter 4 [Pahlavan]
  - Chapter 3 [Garg]

Uplink/Down Link Duplexing

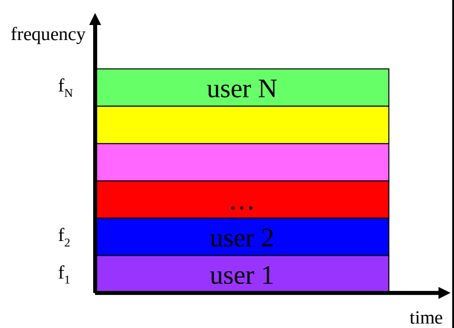
- Mechanism to differentiate between uplink and downlink transmissions
- Two basic techniques are used:
  - Frequency Division Duplexing (FDD)
  - Time Division Duplexing (TDD)
- FDD
  - Usually large coverage areas
- TDD
  - Share one RF circuitry
  - Accurate open-loop power control (refer to IS-95)
  - Usually low-power local communications
- More will be provided later on the pros and cons of each of these technologies

FDMA

- All users may transmit simultaneously – each using a distinct carrier ➔ channel
- Basics: Frequency Division Multiplexing (FDM)
- Design Issues:
  - Adjacent channel interference (refer to backup slides – voiceband signals)
  - RF spectrum mask
  - Near-far problem – a concern especially on reverse link
    - Carriers belonging to one set are not adjacent
    - Guard bands – reduces spectral efficiency

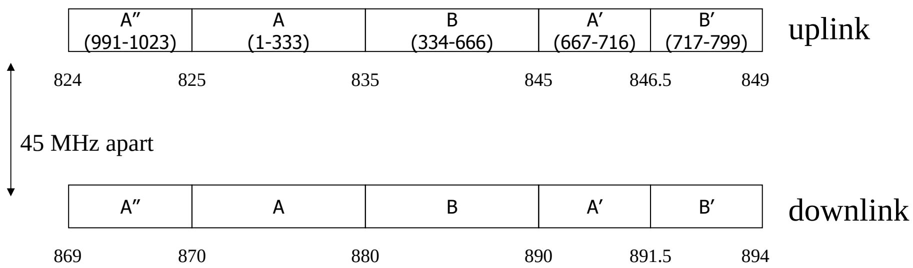
FDMA – cont'd

- A user is assigned a carrier f_i for each direction (uplink and downlink)
- A user may employ continuous transmission
- Data (user's info) is modulated using the assigned carrier
- Analog circuitry (VCO) is required to keep track of frequency shifts



Example: AMPS

- A user is assigned an uplink and a downlink channels that are 45 MHz apart



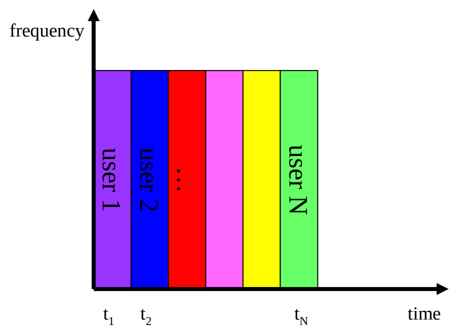
Original FCC designation:
A band for independent carriers
B for traditional wireline (regional Bell Operating Companies) carriers
A', B', and A'' were added later

Time-Division Multiple Access

- A number of users share the same frequency band by taking assigned turns in using the channel
- BS controller assigns slots - slot released upon the completion of call
- Advantages:
  - Flexibility – can provide different access rates at no cost
- Disadvantages:
  - Requires accurate synchronization with BS and rest of users
    - Guard times

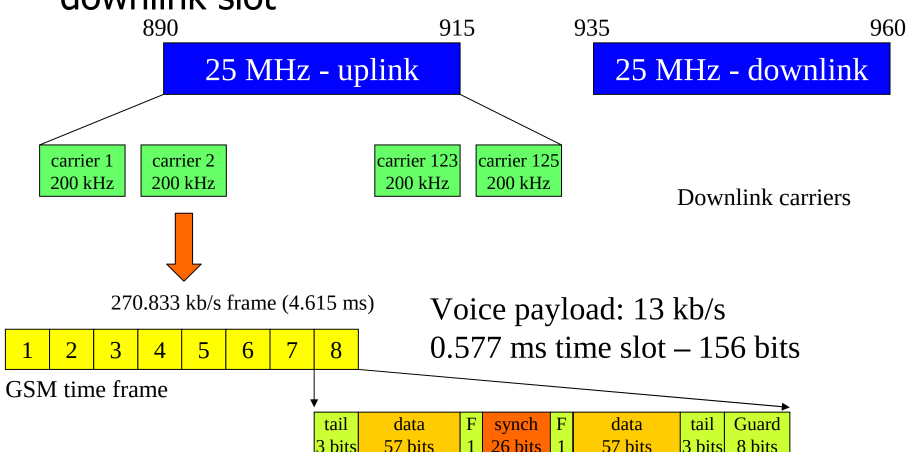
TDMA

- All the band is used by the user during his slot
  - Fixed assignment – predetermined order
  - Slot waster if there is no info for transmission



Example: GSM

- The user is assigned one uplink slot and one downlink slot



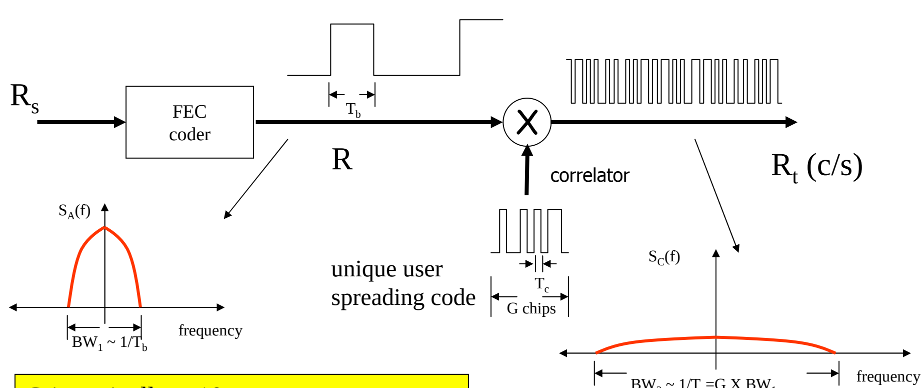
Direct-Spread Code-Division Multiple Access (DS-CDMA)

- Spread Spectrum (SS): a technique in which a signal is of original bandwidth W is transmitted over a bandwidth equal to GXW where G >> 1
  - G is referred to as the spreading gain
- DS-CDMS: A bit stream of rate R b/s (bit duration, T_b = 1/R sec) occupies a bandwidth of W ≈ R Hz
  - The bandwidth is roughly inversely proportional to the duration of the smallest signal element (bit), i.e. W ≈ $1/T_b$ = R
  - If the signal is multiplied by a code: where every bit is multiplied by a sequence of G chips each of duration T_c, then the new signal have a bandwidth ≈ $1/T_c = G/T_b$ = GXW

Frequency hopping (FH) is another form of spread-spectrum technology – review chapter 3 [Pahlavan]

Code-Division Multiple Access (CDMA)

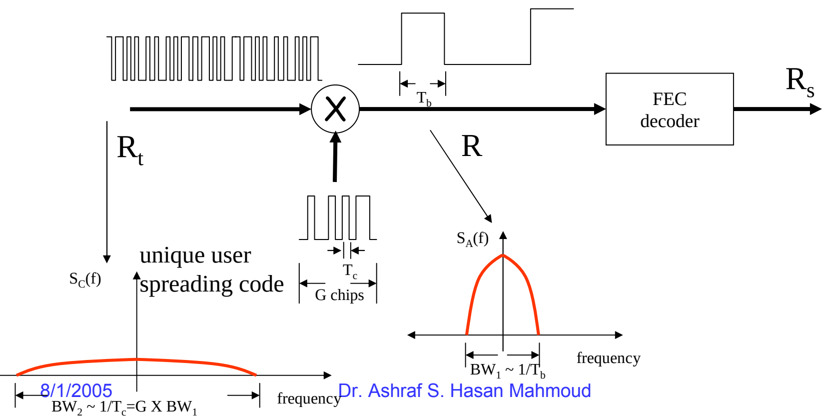
- DS-CDMA Transmitter (simplified)



G is typically > 10
- For IS-95 G = 9.6 kbs/1.25 MHz = 128

Code-Division Multiple Access (CDMA)

- DS-CDMA Receiver (simplified):
  - Typically, the signal is passed through a LPF and then sampled - the samples are then used by the FEC decoder
  - An alternative receiver structure: Replace the correlator by a filter matched to the specific user code



How Does CDMA work?

- Input:
  - Assume we have N users transmitting simultaneously
  - Let the bit stream of the ith user by $R_i(t)$
  - Let the code assigned to the ith user by $C_i(t)$
  - Codes are ORTHOGONAL, i.e.

$$C_i(t) \times C_j(t) = 1 \text{ if } i = j$$
$$= 0 \text{ if } i <> j$$

- Operation:
  - Each user uses its code to spread its signal – the signal transmitted by the ith user is $S_i(t) = R_i(t)C_i(t)$
  - The signal received (say by the base station) is the sum of all transmitted signals (ignore multi-path copies for the time being),

$$S_r(t) = \sum S_i(t) = \sum R_i(t)C_i(t)$$

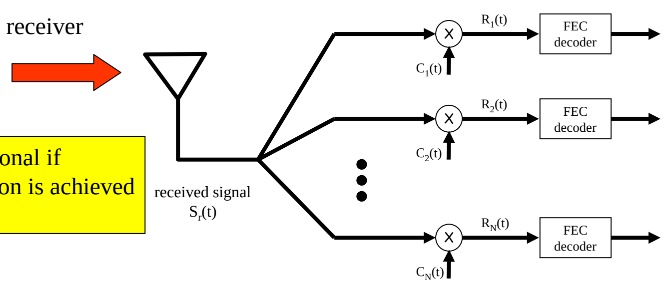
How Does CDMA work? – cont'd

- Demodulation (De-spreading):
  - Receiver dedicates a path structure per user – multiplies the received signal with the kth user code

$$C_k(t) \times S_r(t) = C_k(t) \times \sum S_i(t)$$
$$= C_k(t) \times \sum R_i(t) C_i(t)$$
$$= R_k(t)$$

i.e. only the kth signal is retrieved from the kth receiver path

Simplified basestation receiver



Codes are only orthogonal if
-Perfect synchronization is achieved
-No multipath exists

Code Division Multiple Access

- User transmits all the time (not in a particular slot) and using all the frequency bandwidth
- User is assigned a distinct code
- A frequency reuse factor of one is potentially possible with CDMA