TSG RAN WG1#7 **TSGR1#7(99)b77**
Hanover, Germany
August 31-September 3, 1999

Agenda Item: **AH 14**
Source: **Golden Bridge Technology**
Title: **CPCH Simulations**
Document for: **Decision on use of Idle-AICH and TFCI for CPCH**

Background

GBT has performed extensive simulations of the CPCH using the OPNET Modeler tool. We can provide the source code for other parties independent evaluation. The simulation is a protocol simulation and as such it is a system level CPCH simulation which captures the various impacts such as the traffic model, wireless link quality, mobile distribution and various access methods on the throughput delay performance of the CPCH as proposed by various parties. GBT had presented some analytical results in WG1 in March and these simulations validate those results and extend it further.

The main motivation to present the simulation is to back up our proposal to use idle-random for the CPCH access method. The other motivation is to further justify the use of TFCI.

Simulation Assumptions

1. Results of Link Level Simulations with ITU channel model is used.
2. The preamble detection probability as a function of SNR (Table used).
3. Window-based and timer-based ARQ is used. So we have captured end-to-end delays.
4. 50-200 mobiles are randomly distributed in the coverage area of one cell.
5. The access Preamble ramp-up and the collision resolution steps are simulated.
6. The following tunable parameters exist in the simulations:

N_Max_Frames
Number of ramp-ups
Traffic model, packet inter-arrival time, session inter-arrival time, # of packets per packet call, number of packet calls per session, Session length, average packet size, etc.
New packet model with packet trains
Three various Access Methods

The following traffic model is used in the simulations:

Average packet size: E-mail application 480 bytes
# of packets in a packet call = 15

Packet call inter-arrival time = 120, 20, etc.
# of packet calls within a session =1
Average inter-packet arrival time = 30, 100, 200 ms
Data rates: 2.048 Msps (512 kbps), 384 ksps (96 kbps), 144 ksps (36 kbps), 64 ksps (16 kbps)
Session arrival = Poisson

7. The following results are captured:

End-to-End Delay
RLC queuing delay
Radio Access Delay
MAC collisions
Throughput (S1) includes ARQ re-transmissions/ excludes detected MAC collisions/excludes undetected collisions as well
Throughput (S2) excludes ARQ re-transmissions / excludes MAC collisions
Offered Load (rho)
% First time successful transmissions
Undetected collisions per sec.
Detected collisions per sec.

The three access methods are: Simple, recency, idle-random

A. Simple access method:

In this method, the UE monitors the available capacity and the highest available rate from the Base Node. The UE then picks a slot randomly and contends for the CPCH.

B. The recency table method:

In this method, the UE monitors the AP-AICH and constructs a recency table, which includes time-stamps, which aid the selection of the CPCH channel. The simulation assumes perfect knowledge of the transmission of AP-AICH from the base Node. In reality, there will be discrepancies in the information in the table since the UE is not required to receive FACH, DL-DPCCH and AP-AICH simultaneously.

C. The idle-random method:

In this method, the UE monitors the idle-AICH and AP-AICH and has perfect information on the availability of the CPCH channels. The UE monitors the AP-AICH and CD-AICH for 10 ms. then it picks the CPCH channels randomly from the available ones in the desired data rate category. So, this case is the perfect monitoring case. Note that this method is sensitive to back-off methods. When the traffic load is high and there are multiple CPCH channels, this method outperforms the other methods given the right back-off parameters.

Simulation Results

Cases A-B: Comparison of idle-random method and the recency method for 30 ms packet inter-arrival time, 480 bytes, and 6 CPCH @384 ksps:

We ran over 36 cases to compare the throughput delay performance of the two methods when the packet inter-arrival time is 30 ms. This was done for various packet lengths (158 bytes, 480 bytes, 1000 bytes, 2000 bytes), various rates (6 CPCH @ 384 ksps, 16 CPCH @ 144 ksps, 32 CPCH @ 64 ksps), various N_Max_Frames (8,16,24,32,64), and the three access methods. In all cases, the idle-random method performed better. When the packet inter-arrival time was increased, the throughput delay performance of the recency method almost overlapped with the idle-random case (see Scenarios C-D-E).

Scenario C-D-E: Comparison of the three methods for multiple CPCH

Recency table and the idle random methods out-perform the simple case significantly. However, the recency method performs almost similar to idle-random case in these simulation runs for two reasons: 1) the recency table case in the simulation does not have any discrepancies in its information 2) the back-off for idle-random is not optimized and therefore it performs slightly worse when the packet inter-arrival time is high (e.g., 100 ms).

At D (un) of 300 ms, we have the following throughputs:

1. **Simple case, S1 = .55**
2. **Recency table: S1= .8**
3. **Idle-random S1 = .78**

Cases E-F: Impact of packet inter-arrival time

As we increase the packet inter-arrival time from 100 to 200 ms, the throughput delay performance improves significantly. As we increase the packet inter-arrival time, the packet model resembles the Poisson arrival model more. The motivation to increase the packet inter-arrival time to improve the overall delay performance of all methods. This can be achieved in practice by having the TFCI and being able to send more packets during a single CPCH transmission if it arrives in the RLC buffer. This is quite possible from a single logical channel.

Case G: Number of mobiles in a cell:

There could potentially be hundreds of UEs in parallel session as shown by the table in this case. In case E, there are 920 UEs in parallel session if 25% of the capacity was allocated to Packet Data services.

Cases H-I: Comparison of recency and idle-random methods for single CPCH:

The recency method outperforms the random-idle for a single CPCH case and high inter-arrival time of 200 ms as shown by tables in cases F and G. The reason for this is the non-optimized back-off mechanism for the random-idle case.

Case J and H: Comparison of single CPCH and multiple CPCH, idle-random and 2 Msps:

As can be seen from the table the multiple CPCH case performs significantly better than the single CPCH case. Note that the packet length in the multiple CPCH case is 1000 bytes whereas in the single CPCH case it is 480 bytes. This case outperforms the single CPCH channel with the recency method as well (Case I).

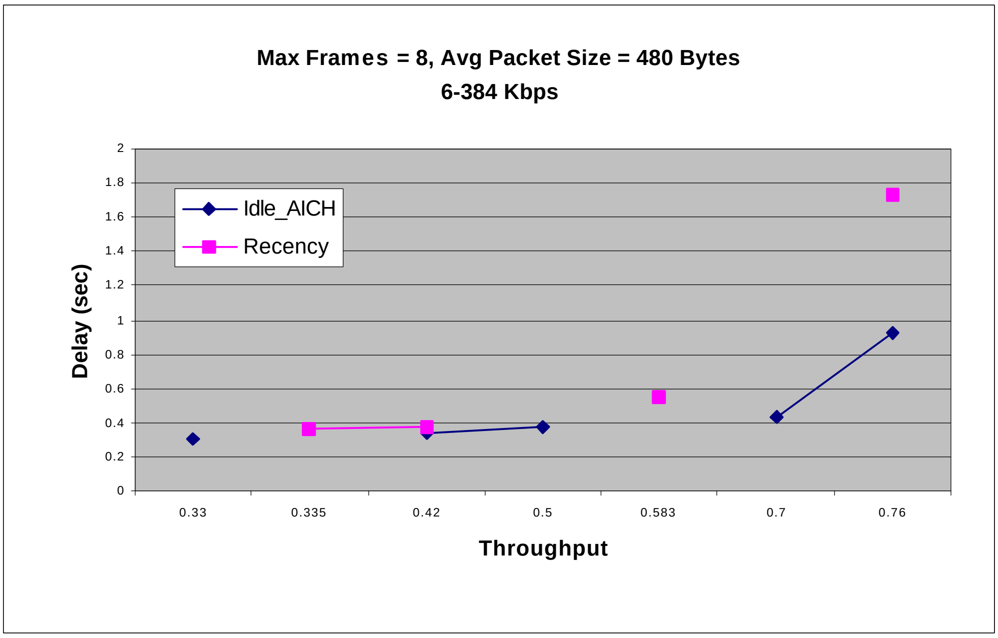
Scenarios A-B: Comparison of idle-random method and the recency method for 30 ms packet inter-arrival time, 480 bytes, and 6 CPCH @384 ksps:

Scenario A: Idle-random case:

ρ	γ	**D(e-e)**
.34	.33	.3
.44	.42	.338
.53	.5	.375
.65	.70	.430
.95	.76	.92

Scenario B: recency table case:

ρ	γ	**D(e-e)**
.36	.335	.36
.45	.42	.375
.67	.583	.55
.97	.76	1.73



Max Frames = 8, Avg Packet Size = 480 Bytes
6-384 Kbps
Delay (sec)
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Idle_AICH
Recency
0.33
0.335
0.42
0.5
0.583
0.7
0.76
Throughput

Scenario C-D-E (Comparison of the three Access Methods)

Packet Inter-arrival time = 100 ms

1) Maximum Frame Per Packet 8

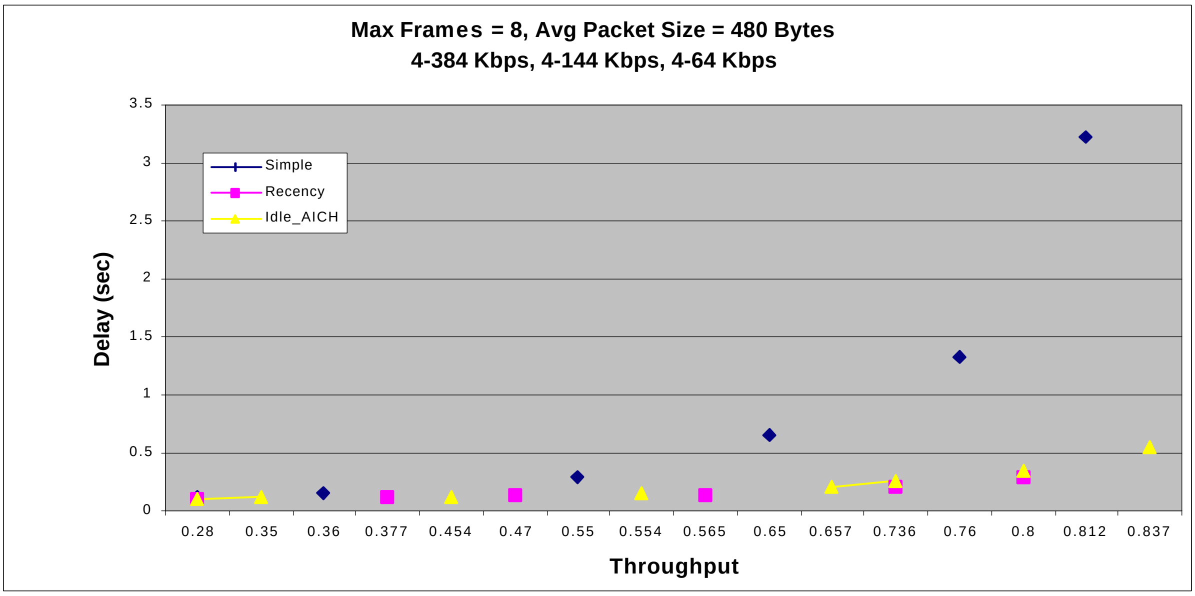
2) Simple Avg Pk Size=480

3) 4-384CPCH , 4-144CPCH , 4-64CPCH

email_1_simple							
Sess	**ρ**	**1S1**	**D(un)**	**QD**	**AD**	**TD**	**MAC Collision**
20	0.310	0.280	0.121	0.070	0.013	0.038	677,000.000
16	0.390	0.360	0.155	0.100	0.015	0.039	106,000.000
10	0.630	0.550	0.300	0.237	0.020	0.042	266,000.000
8	0.776	0.650	0.660	0.589	0.025	0.045	436,700.000
6.8	0.923	0.76	1.324	1.245	0.033	0.046	714,700.000
6.6	**1.00**	**0.812**	**3.23**	**3.15**	**0.036**	**0.047**	**983,300.000**

Email_1_recency							
20	0.283	0.280	0.110	0.062	0.009	0.038	96,500.000
16	0.380	0.377	0.116	0.069	0.010	0.038	162,000.000
12	0.477	0.470	0.131	0.081	0.012	0.038	251,000.000
10	0.566	0.565	0.140	0.088	0.014	0.038	354,700.000
8	0.779	0.736	0.203	0.149	0.016	0.038	733,300.000
7.1	**0.846**	**0.800**	**0.290**	**0.235**	**0.017**	**0.038**	**860,000.000**

Email_1_random							
20	0.282	0.280	0.102	0.056	0.007	0.039	65,100.000
16	0.351	0.350	0.118	0.072	0.007	0.039	89,000.000
12	0.458	0.454	0.124	0.076	0.008	0.040	137,500.000
10	0.558	0.554	0.148	0.109	0.008	0.041	215,000.000
8	0.667	0.657	0.211	0.160	0.009	0.042	344,000.000
7.1	0.741	0.736	0.260	0.208	0.010	0.043	472,000.000
6.5	**0.825**	**0.800**	**0.350**	**0.296**	**0.012**	**0.043**	**644,000.000**
6.3	0.876	0.837	0.544	0.488	0.013	0.043	765,300.000



Max Frames = 8, Avg Packet Size = 480 Bytes
4-384 Kbps, 4-144 Kbps, 4-64 Kbps
Simple
Recency
Idle_AICH
Delay (sec)
3.5
3
2.5
2
1.5
1
0.5
0
0.28
0.35
0.36
0.377
0.454
0.47
0.55
0.554
0.565
0.65
0.657
0.736
0.76
0.8
0.812
0.837
Throughput

Scenario F

Impact of increasing the packet inter-arrival time

Idle-random, 480 bytes, 16 CPCH channels (4 @384 ksps, 4 @ 144 ksps, 4@ 64 ksps)

Packet Inter-arrival value
200

email_3_random							
Sess	**ρ**	**γ**	**D(un)**	**QD**	**AD**	**TD**	**MAC Collision**
20	0.275	0.273	0.067	0.022	0.007	0.038	61,600
16	0.329	0.326	0.074	0.028	0.007	0.039	81,900
10	0.470	0.467	0.076	0.029	0.008	0.040	152,700
8	0.558	0.554	0.079	0.031	0.008	0.041	233,300
7	0.616	0.610	0.091	0.041	0.009	0.042	300,000
6.5	0.656	0.647	0.142	0.091	0.009	0.042	345,300
6.3	0.681	0.673	0.112	0.061	0.009	0.042	388,000
4.95	0.819	0.79	0.178	0.123	0.012	0.043	637,000
4.9	0.867	0.824	0.205	0.148	0.014	0.043	746,700

Scenario G

How many Packet Data terminals in a cell?

25% cell capacity to e-mail application with average packet size of 480 bytes

Packet Inter-arrival time = 200 ms

Mobiles	ρ	γ	D(un)	QD	AD	TD	MAC Coll
318	.257	.256	.08	.031	.011	.038	55,766
750	.609	.604	.137	.078	.017	.042	300,000
930	.798	.772	.241	.175	.022	.044	595,000

930 mobiles in parallel session @ 25% capacity

Scenarios H-I :Comparison of recency and idle-random methods for single CPCH case: 2 Msps, single CPCH, 200 ms packet inter-arrival, 480 bytes messages

Scenario H: Idle-random case

ρ	γ	D(un)	QD	AD	TD	MAC Coll
.56	.535	.23	.171	.0448	.0137	200,833
.768	.684	.97	.883	.0729	.0137	398,000

Scenario I: recency table case

ρ	γ	D(un)	QD	AD	TD	MAC Coll
.574	.634	.0927	.057	.022	.0137	153,333
.813	.675	.131	.086	.031	.0136	318,666

Scenario J: idle-random case: 4 CPCH @ 2Msps, 300 ms inter-arrival time, 100 byte messages

ρ	γ	D(un)	QD	AD	TD	MAC Coll
.57	.61	.067	.02	.012	.035	6.35 %
.76	.71	.096	.045	.016	.035	14.6%
.82	.75	.104	.05	.019	.035	18.1%
.88	.76	.171	.115	.021	.035	20%
.93	.8	.242	.184	.023	.035	23%
.975	.81	.367	.28	.025	.035	25%

Discussion on idle-aich and use of TFCI

As the packet inter-arrival time decreases, the throughput delay performance of the various shades of the CPCH access protocols degrades. At low packet inter-arrival times, the idle-random method clearly out-performs the recency method. The simple method performs worst in all cases. When the packet-inter-arrival time increases to 100-200 ms, then the recency method performs similar to the idle-random case. Note that at high packet inter-arrival times, the throughput delay performance of all cases improves significantly. In reality, if we do not have fixed packet length and let the UE transmit the incoming packets from the higher layer midst the CPCH transmission, then the packet inter-arrival times will be higher values. By optimizing the random-idle case with appropriate back-off mechanism and incorporating the impact of the discrepancies in the recency table, the random-idle case will perform better at high packet inter-arrival times as well. So, we propose adoption of use of idle-aich to provide for more knowledge of the CPCH channel usage.

Recommendations:

Adopt use of idle-AICH to improve the performance when the packet inter-arrival time is small.

Use of TFCI is recommended so that the packet arrival process become less clustered and approach the Poisson statistics. This will ensure better throughput delay performance.