Simulated Performance of W-CDMA Random Access Channel

Prashant Choudhary, Amitava Ghosh and Louay Jalloul

Advanced Radio Technology Group, Motorola

choudh@cig.mot.com, qa0047@email.mot.com, qlj000@email.mot.com

Abstract: *In 3GPP W-CDMA, a Random Access Channel (RACH) is used for random-access transmission from the User Equipment (UE) [1]. The RACH may also be used to carry short packets. The RACH transmission is based on slotted ALOHA approach with fast acquisition indication. In this paper, the performance of the RACH channel with respect to delay and throughput is analyzed for a 19-cell Omni system under fading condition. An adaptive persistence control algorithm for slotted-ALOHA system is also presented.*

Introduction

In 3GPP W-CDMA, a Random Access Channel (RACH) is used for random-access transmission and to carry short packets and control information from the User Equipment (UE) [1] to the Node-B. The RACH transmission is based on slotted ALOHA approach with fast acquisition indication. In Section I of the paper, a brief description of the RACH transmission structure and procedure for W-CDMA system is given. The analytical performance bounds for a slotted ALOHA system is discussed along with the methods to control the throughput of such a system in Section II. An adaptive approach for persistence control is also presented in this section. Section III provides a brief description of the RACH system simulator. In Section IV, the simulated performance of the RACH is studied at various values of vehicle speeds in terms of system throughput and average user access delay. Finally, conclusions are drawn in Section V.

I. RACH Transmission Structure and Procedure

Transmission Structure

The random-access transmission is based on a slotted ALOHA approach with fast acquisition indication. The UE can start the transmission at a number of well-defined time-offsets, relative to the frame boundary of every second frame of the received Broadcast channel (BCH) of the current cell. The different time offsets are denoted as *access slots.* There are 15 access slots per two 10 msec frames and they are spaced 5120 chips apart. Figure 1 shows the access slot numbers and their spacing to each-other. Information on what access slots are available in the current cell is broadcast on the BCH.

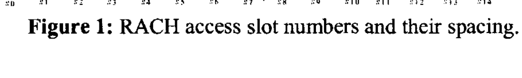

Figure 1: RACH access slot numbers and their spacing.

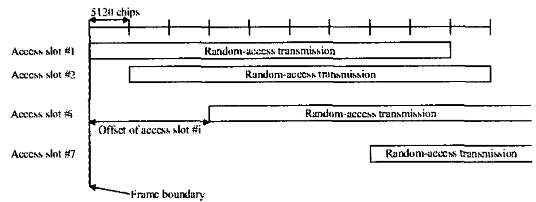

Figure 2: PRACH allocated for RACH access slots.

The structure of the random-access transmission of Figure 2 is shown in Figure 3. The random-access transmission consists of one or several *preambles* of length 4096 chips and a *message* of length 10 ms.

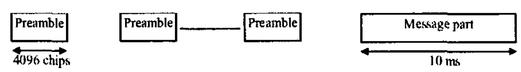

Figure 3: Structure of the random-access transmission.

The preamble part of the random-access burst consists of 256 repetitions of a signature, which is comprised of 16 complex symbols ($\pm 1+j$). There are a total of 16 different signatures, based on the Hadamard code set of length 16. The 10 ms message part is split into 15 slots, each of length $T_{slot} = 2560$ chips. Each slot consists of two parts, a data part that carries Layer 2 information and a control part that carries Layer 1 control information. The data and control parts are transmitted in parallel. The data part consists of $10 * 2^k$ bits, where k=0,1,2,3. This corresponds to a spreading factor of 256, 128, 64, and 32 respectively for the message data part. The control part consists of 8 known pilot bits to support channel estimation for coherent detection and 2 TFCI bits. This corresponds to a spreading factor of 256 for the message control part.

Summary of RACH Procedure

In this paper, the RACH procedure described in [1] is summarized below[1]. Wherever, a random selection is done in the following, a uniform random function is used:

(1) The UE before starting the RACH procedure, receives the preamble scrambling code, message length, transmission timing parameter, available signatures, power ramping factor, maximum allowed retransmissions, initial preamble power, preamble to message power offset and any other information relevant to RACH transmission from the BCH.

[1] The 3GPP specifications have changed considerably since this simulation was done.

(2) The UE implements the dynamic persistence algorithm based on the persistence factor *N*, read from the BCH.

(3) The channel formed by the sequence of access slots, is de-multiplexed into 12 parallel sub-channels and the UE randomly selects one of the available sub-channels for transmission and then randomly selects an access slot in that sub-channel.

(4) The UE randomly selects a preamble spreading code from the set of available codes.

(5) The retransmission counter and the preamble transmission power is initialized.

(6) A preamble is transmitted using the selected uplink slot, preamble signature and preamble transmission power.

(7) If no positive or negative acquisition indicator corresponding to the selected signature is detected in the downlink access slot corresponding to the selected uplink access slot:

  (7.1) A new uplink access slot in the sub-channel is selected.

  (7.2) A new signature from the available signatures is randomly selected.

  (7.3) The preamble transmission power is increased by the ramping factor obtained from BCH.

  (7.4) If the number of transmission attempts is still less than maximum allowed step (6) onwards is repeated. Otherwise RACH procedure is exited.

(8) If a negative acquisition indicator corresponding to the selected signature is detected in the corresponding downlink slot, the RACH procedure is aborted.

(9) If a positive acquisition indicator is detected, the random access message is transmitted three or four uplink access slots after the uplink access slot of the last transmitted preamble depending on the AICH transmission timing parameter.

The dynamic persistence is provided for managing interference and minimizing delay by controlling access to the RACH channel. The system will publish a dynamic persistence value on the BCH, which is dependent on the estimated backlog of users in the system.

II. Adaptive Dynamic Persistence Algorithm

In an uncontrolled slotted-ALOHA system, the rate of arrival of users can be modeled as a Poisson process (with mean P_0) and is independent of *Backlog*. Consequently, the load line (*Rate of Arrival Vs Backlog* Curve) is a straight line, which intersects the slotted-ALOHA performance curve (*Throughput Vs Backlog* Curve) at two points as shown in Figure 4. The points of intersection are the points of equilibrium where the system operates and at these points the *Rate of Arrival* equals the *Throughput*. However, only the point of intersection on the left side of the curve (Point A in Figure 4) is stable, while the point of intersection on the right side is unstable (Point B in Figure 4). This can be seen intuitively because if the *Backlog* goes up at point A, the *Throughput* also goes up, which reduces the *Backlog*. But at point B, if the *Backlog* goes up, the *Throughput* goes down and the *Backlog* will continue to increase without bound. Thus in an uncontrolled system, the point of operation may suddenly jump from point A to point B resulting in an infinite *Backlog*. In controlled slotted-ALOHA system, the *Rate of Arrival* of users is made a function of *Backlog* by some persistence control technique, such that the modified load line preferably, has only a single point of intersection on the left side of the system performance curve. Figure 4 shows some examples of stable and unstable load-lines. Among different load functions, exponential, step and μ-law have been well studied and known to give a stable throughput [5].

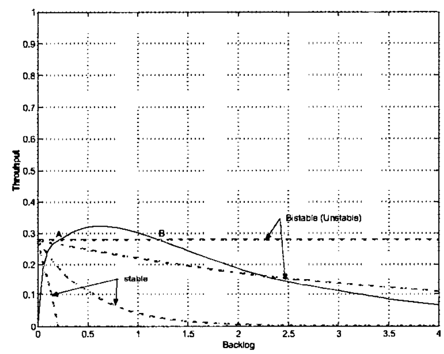

Figure 4: Examples of load-lines, which give stable and bi-stable (unstable) performance.

Therefore, if the performance curve is known *a priori*, along with the mean Poisson arrival rate, a load line can be designed by choosing appropriate parameters, which maximizes the throughput while keeping the system stable. But in a mobile wireless environment, the performance curve is non-stationary and the mean Poisson arrival rate, P_0 also varies from time to time. Hence the parameters of a load line cannot be chosen *a priori*. In the following, we describe a technique in which the load curve parameters adapt to the changes in the system performance curve as well as the Poisson arrival rate:

Key Notations and Assumptions:

- $\widehat{B}(n)$ -Estimate of User Backlog in frame *n*
- $\widehat{\overline{B}}(n)$ -Estimate of average Backlog after frame *n*
- $T(n)$ -Throughput in frame *n*
- $\overline{T}(n)$ -Average Throughput after frame *n*

- K_p -Parameter, which controls the location of load line on the performance curve
- $\Delta_{Up} / \Delta_{Dn}$ -Perturbation on K_p to move the load line up/down
- δ_1 / δ_2 -Fixed parameters to control the shape of the load line
- $P_0(n, \hat{B}(n))$ -Effective rate of arrival of users in the nth frame
- $p(n, \hat{B}(n))$ -Persistence factor for frame n (the natural rate of arrival is reduced by this factor)
- α, β -Parameters for averaging $\hat{B}(n)$ and $T(n)$

The persistence factor is determined by the following equation:

$$p(n,\hat{B}(n))=\begin{cases} e^{-\left(\hat{B}(n)/K_p\right)} & (0 \le \hat{B}(n) < \hat{\bar{B}}(n)+\delta_1) \\ e^{-\left(\hat{B}(n)+\delta_1/K_p\right)} * \left(\dfrac{\hat{\bar{B}}(n)+\delta_1+\delta_2-\hat{B}(n)}{\delta_2}\right) & (\hat{\bar{B}}(n)+\delta_1 \le \hat{B}(n) < \hat{\bar{B}}(n)+\delta_1+\delta_2) \\ 0 & (\hat{\bar{B}}(n)+\delta_1+\delta_2 \le \hat{B}(n) < \infty) \end{cases}$$

Therefore, if P_0 is the natural Poisson rate of arrival of UE packets, the modified rate will be:

$$P_0(n, \hat{B}(n)) = P_0 \cdot p(n, \hat{B}(n))$$

This is the equation of the load line as a function of Backlog in frame *n*. Figure 5 shows different load curves for different values of K_p. The shape of the load line is a pseudo-step function (partly exponential and partly step), and it will always intersect the performance curve at a single point. As the value of K_p increases the load line moves further up on the performance curve. For stability the intersection point should lie on the left side of the performance curve. Further, to maximize the throughput it should be as far up on the left side as possible. The slope of the performance curve at the point of intersection with the load line can be used to find out the current location of point of operation and the load line can be modified accordingly. This is the key idea in the following algorithm, which keeps shifting the load line on the performance curve till it reaches the maxima of the curve. The algorithm proceeds as follows:

1. After every M frames, the expected slope at the current point of operation (intersection of present load line and performance curve) is computed. If the slope is "+ve", $K_p = K_p + \Delta_{Up}$, which moves the load line further up on the performance curve. If the slope is "-ve", $K_p = K_p - \Delta_{Dn}$, which brings down the load line (Figure 5). Note that the step function like load line also gives a bit of stability to the system because if the backlog becomes too large suddenly, the user arrival rate drops rapidly to zero preventing any catastrophic backlog.
2. Since only the sign of the slope is important, it is computed just by comparing the changes in $\bar{B}(n)$ and $\bar{T}(n)$ every M frame. Therefore,
   - Slope = "+ve" if $\hat{\bar{B}}(n) - \hat{\bar{B}}(n-M) > 0$ when $\bar{T}(n) - \bar{T}(n-M) > 0$ & $\hat{\bar{B}}(n) - \hat{\bar{B}}(n-M) < 0$ when $\bar{T}(n) - \bar{T}(n-M) < 0$
   - Slope = "-ve" if $\hat{\bar{B}}(n) - \hat{\bar{B}}(n-M) > 0$ when $\bar{T}(n) - \bar{T}(n-M) < 0$ & $\hat{\bar{B}}(n) - \hat{\bar{B}}(n-M) < 0$ when $\bar{T}(n) - \bar{T}(n-M) > 0$
3. $\hat{\bar{B}}(n)$ and $\bar{T}(n)$ are computed as follows:

$$\hat{\bar{B}}(n) = (1-\alpha)\hat{\bar{B}}(n) + \alpha\hat{B}(n)$$
$$\bar{T}(n) = (1-\beta)\bar{T}(n) + \beta T(n)$$

III. Simulator Description

Figure 9 shows the block diagram of the simulator. The traffic generator block generates arrivals according to a Poisson process every two frame (i.e. every 20 msec). When an arrival is generated it is assigned an access slot (a number between 1 and 15) and a preamble sequence (a number between 1 and 16). The number of new users admitted into the system is reduced by a Persistence factor $p(n, \hat{B}(n))$, as computed in Section II. It may be noted that new users, not admitted to the system due to persistence, are dropped out of the system. The users (new + backlogged) are then dropped randomly in the center cell of a 19 hexagonal cell Omni-system. The user locations are uniformly distributed over the cell area. The attenuation between a UE and the i^{th} cell site is modeled by

$$L_i = D_i^{-\mu} 10^{X_i/10} R_i^2$$

where D_i is the distance between the UE and the cell site, μ is the path loss exponent and X_i represents the shadow fading which is modeled as a Gaussian distributed random variable with zero mean and standard deviation σ. X_i may be expressed as the weighted sum of a component Z common to all cell sites and a component Z_i which is independent from one cell site to the next. Both components are assumed to be Gaussian distributed random variables with zero mean and standard deviation σ independent from each other, so that

$$X_i = aZ + bZ_i \text{ such that } a^2 + b^2 = 1$$

And R_i^2 is the Rayleigh fading component. Typical parameters are $\sigma = 8$ and $a^2 = b^2 = 1/2$ for 50% correlation. The UE selects the cell with the largest received Pilot power, i.e. the minimum loss. It is assumed that all cell sites are transmitting at the same peak power. The UE transmitted power to its corresponding cell site is then computed subject to a maximum transmitted power limit of 200 mW. It may be noted that for backlogged UE's accessing the system the transmitted power is increased by a fixed delta after each retry. The total interference in a slot is computed by summing up the transmit power from each UE in a particular slot. The received Ec/No at the BTS for the preamble is then calculated. The corresponding Eb/No is then calculated by averaging the Rayleigh samples over 8 slots. To account for other interfering user services a noise rise of 3 dB is added.

In the next step some UE packets are backlogged based on the computed Ec/No and Eb/No. A P_D (Probability of preamble detection) vs. Ec/No curve (for the preamble part) and FER (Frame Error Rate) vs. Eb/No curve (for the message part), for an AWGN channel, is created using the link level simulator. These curves are used to create a mapping from Ec/No and Eb/No to the corresponding P_D and FER. To determine whether a preamble is backlogged or not, a uniform random number (between 0 and 1) is generated and compared with P_D. If it is greater than P_D, the UE is backlogged. A similar procedure is also used for the message part. Some more UE packets are backlogged if they collide with each other. A packet is marked collided, a) if there are two or more packets with the same time-offset and the same preamble sequence and b) if the packets with the same time offset and preamble sequence, differ in power level by more than a fixed threshold denoted as the *Capture Ratio*. Delay is computed for each backlogged user. It may be noted that the maximum number of retransmissions is limited to 10.

IV. Simulation Results

Simulations were run with the RACH simulator for various values of Erlang load with and without Dynamic Persistence enabled, at vehicle speeds of 0, 3, 20 & 60 mph respectively. The parameters used in the simulation are listed in Table 1. Figure 6 shows the *Throughput Vs Backlog* performance curve obtained at various speeds when there was no capture effect (*Cap Ratio = infinity*). The *Throughput* and *Backlog* is given in *packets/slot* where each pair of time-offset and preamble code comprise a slot. Since there are 16 code sequences and 15 slots, *Throughput* (or *Backlog*) in *packets/slot* should be multiplied by a factor of 240 (16×15) to get *Throughput* (or *Backlog*) in terms of number of users. As an example, *Throughput* of 0.37 *packets/slot* corresponds to approximately 90 users per 20 msec frame. The performance curves for non-zero speeds show similar behavior because there is no interleaving effect. Figure 7 shows the performance curve with *Capture Ratio* = 10 dB. For non-zero speed, the peak *Throughput* increases to 0.33 compared to 0.27 for the no capture case, while for the static case the change is insignificant. Figure 8 shows the *Delay Vs Backlog* performance of RACH at different speeds. In the region of stable operation the average delay for backlogged users, which finally get through, is between 20 msec to 60 msec. The adaptive dynamic persistence algorithm provides 90-95% of maximum possible throughput.

V. Conclusions

The following conclusions are drawn from the simulation:

1. *Capture Effect* improves the performance of slotted-ALOHA in a fading environment by 20%.
2. The RACH can support 60 to 80 users per 20 msec (two radio frames) with an average backlog delay of between 20 msec to 40 msec, in the region of stable operation
3. The adaptive Dynamic persistence algorithm gives a good approach to control the throughput of slotted-ALOHA systems without having to design the load lines for a particular performance characteristic and fixed arrival rate.

References

[1] 3GPP RAN WG # 1, Technical Specification Group (TSG) 25.211-25.214, V3.1.1.

[2] *K. Joseph and D. Raychaudhuri*, "Analysis of Generalized Retransmission Backoff Policies for Slotted-ALOHA Multi-access Channels", *IEEE Trans on Comm.* Vol.88, Jan. 1990, pp.117-122.

[3] *D. Covarriubias, S. Ruiz, J. Huguet and J. Olmos*, "Spatial Distribution Analysis with Capture Effect of a Mobile S-ALOHA Network", *IEEE PIMRC* 1998, pp.1116-1120.

[4] *D. G. Jeong and W. S. Jeon*, "Performance of an Exponential Backoff Scheme for Slotted-ALOHA Protocol in Local Wireless Environment", *IEEE VTC* 1995, pp.470-479.

Table 1: Simulation Parameter List

Parameter	**Value**
Number of Cells	19
Antenna Pattern	Omni-directional
Path Loss exponent	4 (i.e. 40log10(distance)
Log-normal Standard deviation	8dB
Fast Fading Channel Model	1-path, Jakes Spectrum
Site-to-Site Correlation	50%
User Arrival Distribution	Poisson
Cell Geometry	Hexagonal
User location placement	Uniform over area
Mobile speed	Variable
Arrival rate	Variable

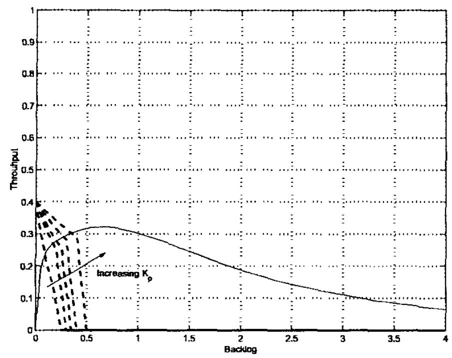

Figure 5. Load Line moves up as K_p increases.

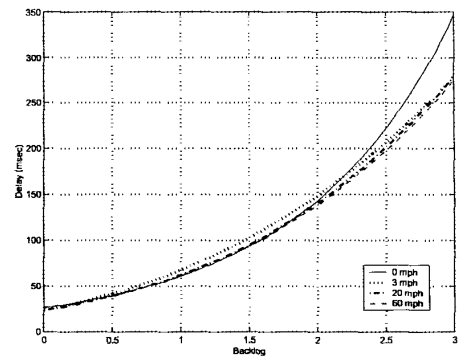

Figure 8: Delay Vs Backlog performance of RACH

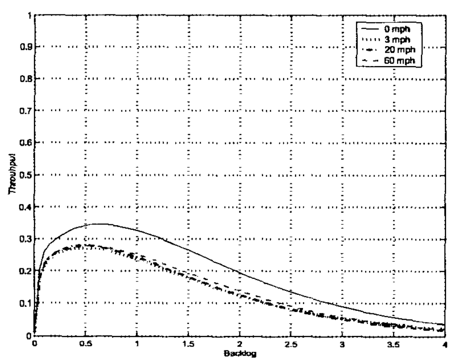

Figure 6: Throughput Vs Backlog performance for RACH channel with no capture

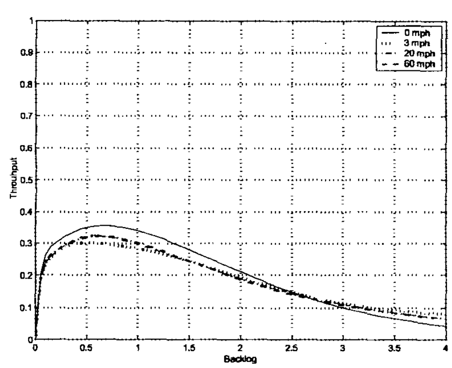

Figure 7: Throughput Vs Backlog performance for RACH channel with *cap ratio* = 10 dB

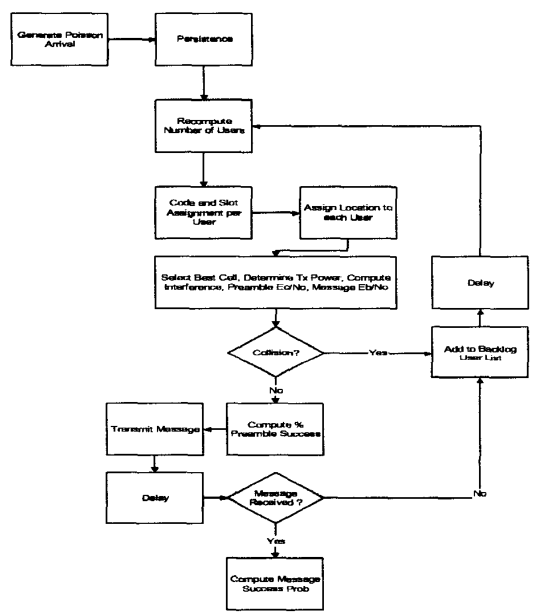

Figure 9: Simulator Block Diagram