Delay Analysis of Downlink IP Traffic on UMTS Mobile Networks

Jesse Landman and Pieter Kritzinger

Data Network Architectures Group
Computer Science Department
University of Cape Town, South Africa
e-mail: {jlandman,psk}@cs.uct.ac.za

Abstract

Since wireless networks which can carry high bit rates have become ubiquitous, mobile computing is no longer just spoken about. Mobile computing always implies access through a wireless network to an IP network such as the Internet. In order to understand the performance of such links, we propose an analytic model for the down link delay of IP traffic between the Mobile Gateway Server and the End User in a UMTS mobile network. Traffic arriving at the Gateway Server is considered to be bursty in nature and we use a Batch Markovian Arrival Process (BMAP) to model this arrival process. We model the wireless link itself as a modified multi-state Gilbert-Elliot Markov model which takes into account the number of interfering users and whether the channel experiences Ricean fading or not for what we consider a typical indoor, IP-centric environment. We also account in both the analytical model and the simulation for the Forward Error Correction provided by Turbo coding in UMTS to establish realistic packet retransmission rates. Finally we calibrate and verify the correctness of the model with a discrete event simulator.

Key words: Mobile computing, IP Traffic, BMAP, WCDMA, Turbo Coding, Ricean Fading, BMAP/D/1 queue

1 Introduction

With the increasing usage of mobile computing the performance of the radio link in the path from server to client in the fixed to wireless network path is receiving increasing attention. The mobile computing traffic is invariably IP based, while the physical network itself is generally either WiFi (IEEE 802.11n) or UMTS. UMTS is different in that it is a much longer range network and thus subject to signal *fading*, which arises as a results of multiple versions of a signal arriving with various amplitudes and phases at the receiver due to scattering. Two kinds of fading environments are known: Non-Line-of-Sight (NLOS), where the fading signal is approximated by Rayleigh distributions, and signals with a dominant Line-of-Sight (LOS) component, where the fading signal is approximated by Ricean distributions. The reader unfamiliar with UMTS and the terminology concerned is referred to one of several good textbooks such as that by Rappaport[20].

WiFi performance has been analyzed by several authors such as [13, Johnson] and is not the subject of this paper. We concern ourselves with a model to analyze the performance of the radio down link (UTRA) of a UMTS network. In particular, we consider the delay path of an IP packet from its arrival at the Gateway GPRS Support Node(GGSN), passing through the Serving GPRS Support Node(SGSN)to the Radio Network Controller (RNC) and its ultimate transmission by the relevant Node B of the UTRAN. This transmission path is illustrated in Figure 1. The service time at Node B is a deterministic

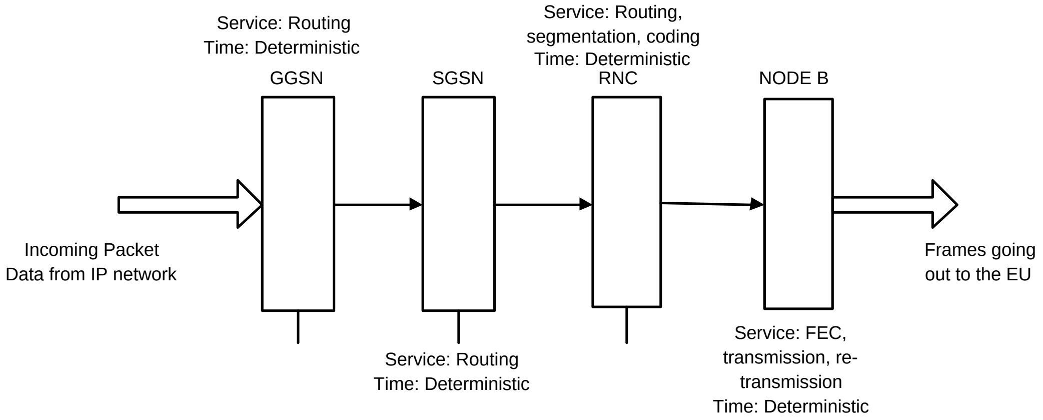

Fig. 1. Conceptual Universal Terrestrial Radio Access (UTRA) component of UMTS

Wideband Code Division Multiple Access (WCDMA) process, but the error conditions on the wireless link is a complex function of the fading process and the Multiple Access Interference (MAI) caused by other users. UMTS links use a Forward Error Control (FEC) coding technique known as Turbo-coding and re-transmission of physical transport blocks occur only when the FEC fails.

Several authors such as [3, 14, 18, 21, 7] have taken up the challenge of modeling wireless channels and the effect of Rayleigh or Ricean fading on the transmission errors. In most cases the models are for flat fading channels where the received signal is a function of the signal power and time-variant interference noise. A Hidden Markov Model (HHM) was presented by Judge and Takawira [14] which extended the Gilbert-Elliot models [19] to include the case where both the user signal power and the interference signal power are varying as is typically the case for WCDMA transmissions.

Several other authors [19, Pimental and Blake], [17, Milstein *et al*] have used HMM's to model wireless channels. In particular, Turin and van Nobelen[21] provides a concise overview of HMM's as a technique for modeling fading wireless channels. Earlier work by Geraniotis[9, 8] lays the foundation for many of the papers.

We generalize the result of [14] for Rayleigh faded channels to those for Ricean fading, introduce a somewhat more general traffic model and represent Turbo coding in our analysis. We have moreover been unable to find any analysis which combine the delay at Node B with the delays at the other points along the downward path from the IP network to the mobile end user as illustrated in Figure 1.

The layout of the paper is the following: In the next section we describe the processing in the GGSN and SGSN modules in more detail and the queueing models used to derive the delay time. IP traffic is modeled by a Batch Markov Modulated Arrival Process (BMAP) as described in Section 2. The major part of the paper then follows, describing the model of the wireless link in Section 3 including the FEC process implemented by Turbo coding in our example and described in Section 3.5. In the final section we use a discrete event simulator to calibrate certain parameters of the analytic model and then proceed to verify the analytic work using a discrete event simulator.

2 End-to-end Model

Figure 1 illustrates that IP packets pass through the GGSN and the SGSN before being processed at the RCN. Both these nodes are effectively routers involving comparatively little delay. In what follows we therefore ignore these delays which imply that IP packets effectively arrive at the RCN for processing. The processing therefore reduces to the processing in the RNC and Figure 2 schematically illustrates the delays involved. At point a in that figure, the arrival process is assumed to be Poisson. Processing occurs at points b and c before the segmented batches of transport blocks arrive at the Media Access Control (MAC) queue for scheduling. The amount of processing required to compress a small IP header and segment a packet into separate data blocks, is much larger than the amount of processing required to encode transport blocks with their channelization and scrambling codes. We therefore assume that the effect of the PDCP and RLC processing in Figure 2 is negligible in terms of time spent, and thus can assume that the arrival process at the Gateway translates to arrivals at the MAC queue as a batch Poisson process, as each IP packet which arrives into the RLC layer is segmented into transport

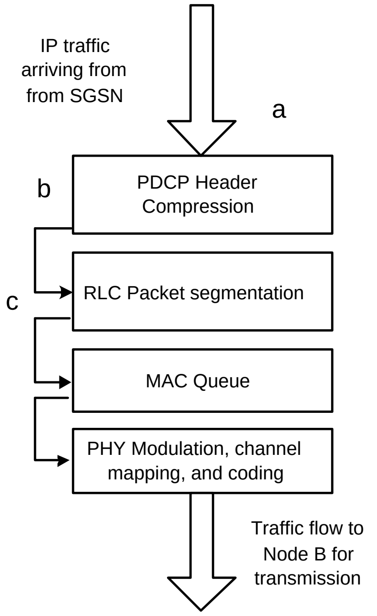
IP traffic arriving from from SGSN
a
b
PDCP Header Compression
c
RLC Packet segmentation
MAC Queue
PHY Modulation, channel mapping, and coding
Traffic flow to Node B for transmission

Fig. 2. Processing delays in the RCN

blocks.

Although we assume that the IP packet arrival process is Poisson, it is clear from what has gone before that every packet yields link layer transport blocks transmitted as physical blocks fitted into a UTRA Downlink Shared Channel (DSCH) frame every 10 milliseconds. The number of transport blocks clearly depend on the arrival rate and the packet size and the process is clearly not a simple Poisson one but rather bursty in nature. The Batch Markovian Arrival Process (BMAP)[1] provides a far more accurate view of IP traffic, because it captures two important statistical properties of IP traffic, namely *self-similarity* and *burstiness*. With this assumption the delays in the RCN described above can be modeled by a BMAP/D/1 queue.

The BMAP traffic arrival process is difficult to use analytically, and its important metrics are often calculated numerically and requires a non-trivial parametrization when fitted to measured data as mentioned in Section 6. The distribution for the time in the BMAP/D/1 queue can be found in [5, Lucantoni] and is summarized very briefly here.

Define $\bar{\mathbf{W}}(x) = (\bar{W}_1(x), ..., \bar{W}_m(x))$, where $\bar{W}_j(x)$ is the joint probability that the arrival process is in phase $j = 1, \ldots m$ and a customer waits at most for a time x before entering service. If the Laplace-Stieltjes transform of $\bar{\mathbf{W}}(\mathbf{x})$ is $\mathbf{W}(s) = \int\limits_0^\infty e^{-sx} d\bar{\mathbf{W}}(x)$, then

$$\mathbf{W}(s) = s(1-\rho)\mathbf{g}[s\mathbf{I} + D(h(s))]^{-1}\,, \quad \mathbf{W}(0) = \pi \tag{1}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{h}(\mathbf{s})$ is the transform of the arbitrary service time distribution $H(x)$, ρ the traffic intensity and $\mathbf{g}$ is a rather complex function of the *busy period*[5].

According to [4, Lucantoni], the waiting time distributions is given by

$$\bar{w}(x) = \bar{\mathbf{W}}(x)\mathbf{e} \tag{2}$$

so that the mean time W in the queue is given by

$$\bar{W} = \int\limits_{x=0}^{\infty} x\bar{w}(x)dx. \tag{3}$$

Several methods exist for solving Equation 1. These methods are fairly complicated and involve many separate techniques and theories. For example, the methods described by Lucantoni[4] draw on several sources, and require methods such as those defined by Abate and Whitt[12] in order to invert the Laplace-Stieltjes transform in Equation 1.

Rather than solve the analytic equations for the waiting time, we simulated the queue using the BMAP parameterized with real data using a software package `IP2BMAP` and provided by [16, Lindemann *et al*]. The algorithms used are described in detail by [2, Klemm].

3 Modeling the Wireless Link

The main contribution of this paper is an analysis of the processing at Node B and its effect on the end-to-end performance model. The Downlink Shared Channel in UTRAN is carried by a radio frame with 15 slots transmitted every 10 milliseconds. A physical block fits into one of the slots and is of fixed size so that the service time at Node B is deterministic. The purpose of the analysis to follow is to model the physical channel and its performance as a function of the Spreading Factor (SF)[20], the error rate and the effect of the Turbo coded error correction. All of these imply that a single IP packet is segmented into several physical blocks for transmission, that overhead parity blocks are generated and that all may be transmitted more than once, thus contributing to the block and overall packet throughput.

The physical wireless channels used by UMTS mobile networks are defined by the multiplexing technique known as Wideband Code Division Multiple Access (WCDMA). WCDMA systems share the transmission medium by allowing all signals to be transmitted simultaneously and at any time. This is achieved by multiplying each signal by a unique, orthogonal spreading code. Due to the multi-user nature of mobile systems, the transmitted signals can be affected by *Multiple Access Interference* or MAI. This occurs when multiple, non-orthogonal signals are transmitted simultaneously. MAI is worsened by *multipath fading* which is a signal distortion that is characteristic of any radio transmission channel. Fading causes multiple version of a signal to exist, usually differing in amplitude out of phase with the original, which can render the orthogonality useless.

It is important to consider these factors when modeling a network that is based on such technology. The following sections detail the model that we have developed for the purpose of analyzing a WCDMA radio channel on an indoor, downlink, shared UTRAN channel that experiences Ricean fading. Before embarking on that analysis however, we define the performance metrics and related quantities that we shall use.

3.1 Performance metrics

The ultimate performance metric of our model is the average IP packet throughput or delay. In order to compute this we need to know the probability that all slots belonging to the packet are received without error by the mobile unit. The success of a slot depends on the state of the network during its attempted transmission and the number of other users present. The following performance metrics are similar to that used by [14, Judge].

We define state x as the number of transmitting users in a time slot and $P\{x = j\}$ as the probability of being in state j. According to our Poisson assumption for the interfering user arrival process,

$$P\{x = j\} = \sum_{i=0}^{\infty} P\{x = i\}\pi_{ij} \quad \text{and} \quad \sum_{j=0}^{\infty} P\{x = j\} = 1 \tag{4}$$

where π_{ij} is the steady state probability of moving from state i to state j and is given by

$$\pi_{ij} = \begin{cases} \sum\limits_{k=0}^{i} \begin{pmatrix} i \\ k \end{pmatrix} \cdot \nu^k \cdot (1-\nu)^{i-k} \\ \qquad \cdot f(j-i+k) \qquad i \leq j \\ \\ \sum\limits_{k=(i-j)}^{i} \begin{pmatrix} i \\ k \end{pmatrix} \cdot \nu^k \cdot (1-\nu)^{i-k} \\ \qquad \cdot f(j-i+k) \qquad i > j \end{cases}$$

where $f(m)$ is the Poisson distribution and the number $L(k)$ of slots in a packet is given by the geometric distribution

$$L(k) = \nu^k \cdot (1-\nu)^{i-k}. \tag{5}$$

We chose $\nu = 3$ for the experimental results given in Section 6.

In order to observe the error sequence process, we write α_{ij} as the joint conditional probability of the successful transmission of the n^{th} and the $(n-1)^{th}$ slots in the packet conditioned on there being j users present during the transmission of the n^{th} slot while there were i simultaneous users during the transmission of the $(n-1)^{th}$ slot.

To complement α_{ij}, we write α_{0j} as the probability of success of the first slot in the packet conditioned on there being j users in the channel at the time of transmission of this first slot. These two definitions are used in $R_n(j)$, which is the probability of success of all slots in a packet up to and including the n^{th} slot given that the n^{th} slot sees j other transmitting users. This probability is solved recursively using

$$R_1(j) = P\{x = j\}\alpha_{0j} \tag{6}$$

$$R_n(j) = \sum_{i=1}^{\infty} R_{n-1}(i)\pi_{ij}\alpha_{ij} \quad n > 1 \tag{7}$$

The probability of success of a packet containing l slots, R_l, is given by:

$$R_l = \sum_{j=0}^{\infty} R_l(j). \tag{8}$$

Assume that it requires $N_x(k)$ transmissions (including the first) to send a packet without error. That is, $k-1$ blocks experience an error and the last block does not. The average number of transmissions for a packet of length L will thus be

$$\bar{N}_x(L) = \sum_{l=1}^{L} \sum_{k=1}^{\infty} k R_l [1 - R_{l-1}]^{k-1}. \tag{9}$$

Finally then the average time $\bar{T}_s$ spent by a packet from arriving at the GGSN until successfully received by the mobile End User (EU) is given by

$$\bar{T}_s = \bar{N}_x(L) W \tag{10}$$

where W is the mean time in the BMAP/D/1 queue given by Eq. 3.

In the following sections we describe how we computed the various quantities used in this section.

3.2 Hidden Markov Model

Hidden Markov Models (HMMs) can be defined as stochastic, finite-state automata that consist of a set of finite states $\mathbf{Q} = \{q_1, q_2, .., q_n\}$, each of which is associated with a specific probability distribution[10]. A transitions from states q_i to state q_j occurs with probability p_{ij}.

Hidden Markov models have been used extensively to model radio channels because of the need to model the *memory* inherent in radio channels; this memory is introduced by the fact that errors often occur in bursts, and are thus statistically dependent [19]. Given a channel input x_k at time interval k, and given that the Markov chain was in state s_{k-1} during the previous interval, we can determine the output of the channel, y_k, from the conditional probability $P(y_k, s_k | x_k, s_{k-1})$. The states of the HMM are associated with channel conditions, while the difference between y_k and x_k is defined as the error sequence. Thus, the HMM is capable of modeling the error sequence and thus the channel memory, despite the fact that it has as a major component a Markov chain, which is defined as being memoryless.

The HMM model we use is adapted from that used by Judge and Takawira [14] who in turn extended a popular HMM known as the Gilbert-Elliot Channel (GEC)[6]. Judge and Takawira extend the GEC model by creating N *good* states and N *bad* states rather than use only two states. Each state is defined as being either *good* or *bad*, conditioned on the number of transmitting users, j, $0 \leq j \leq N$. The probability of being in state $\Omega \in \{good, bad\})$ with j interfering users is $P_j(\Omega)$ in this way accounting for the interference of other users. Bit errors occur in the *Bad* state with probability *h(j)* and in the *Good* state with probability *k(j)* where $0 \leq k(j) \leq h(j) \leq 1 \forall j$. Note that even a single bit error detected by the CRC procedure, unless it can be corrected through the FEC Turbo coding, renders a block in error. In what follows we assume that $k(j) = 0, \forall j$, that is, no errors occur when the channel is in a good state.

The model has many states described by $\mathit{\Omega}_n$ where $\Omega_n = \{G_j, B_j\}$ i.e. G_j denotes being is a *Good* state with j transmitting users, $j = 1, \ldots$ at timeslot n, and B_j denotes being is a *Bad* state with j transmitting users, $j = 1, \ldots$ at timeslot n. $P(G_j)$ denotes the probability of being in this state and ω_{ij}^{CD} as the probability of moving from state C_i to state D_j with $\{C_i, D_j\} \in \Omega_n \forall i, j$. Then

$$\begin{aligned} P(G_j) &= \sum_{i=0}^{\infty} P(G_i)\omega_{ij}^{GG} + \sum_{i=0}^{\infty} P(B_i)\omega_{ij}^{BG} \\ P(B_j) &= \sum_{i=0}^{\infty} P(G_i)\omega_{ij}^{GB} + \sum_{i=0}^{\infty} P(B_i)\omega_{ij}^{BB} \end{aligned}$$

where clearly

$$\sum_{i=0}^{\infty}[P(G_i) + P(B_i)] = 1 \tag{11}$$

Using Bayes' theorem we can write

$$P(C_j) = P\{\Omega_n = C|x = j\} \cdot P\{x = j\} \tag{12}$$

and writing $P\{x_{n+1} = j|x_n = i\}$ as the steady state probability, given by Eq. 5, of moving from state i to j we have

$$\begin{aligned} \omega_{ij}^{CD} = & P\{\Omega_{n+1} = D|\Omega_n = C, x_{n+1} = j, x_n = i\} \\ & \cdot P\{x_{n+1} = j|x_n = i\} \end{aligned} \tag{13}$$

$P\{\Omega_{n+1} = D|\Omega_n = C, x_{n+1} = j, x_n = i\}$ is the steady state transition probability for being in state Ω_{n+1}.

With the above we can now return to Eq. 6 and write

$$\alpha_{0j} = P\{\Omega_n = G|x_n = j\} + P\{\Omega_n = B|x_n = j\}(1 - h(j)) \tag{14}$$

and

$$\alpha_{ij} = P\{\Omega_{n+1} = G|\Omega_n = G, x_{n+1} = j, x_n = i\} \tag{15}$$

The α_{ij} define our error processes i.e., the probability of a single slot being transmitted without error in Eq. 14, and the probability that a slot will succeed given that the previous slot succeeded as well.

For the computation of the probabilities in Eq. 14 and Eq. 15, we need to know the conditional PDFs for the MAI and user signal amplitudes, given by $P_{MAI}(y|j)$ and $P_{sig}(u)$ respectively, since the probability of being in a faded channel depends on the relative amplitudes of the user signal and the MAI. To determine these, we define a parameter θ to be the MAI signal to user signal ratio and the threshold which determines the channel state:

$$y/u \leq \theta \Leftrightarrow \Omega_n = Good \ , \ \ y/u > \theta \Leftrightarrow \Omega_n = Bad \tag{16}$$

Using θ we can now write

$$P\{\Omega_n = G|x_n = j\} = \int_0^\infty \int_{\frac{y}{\theta}}^\infty P_{MAI}(y|j).P_{sig}(u) du dy \tag{17}$$

Note that $P\{\Omega_n = C|x_n = j\}$ is defined as the sum of all instantaneous amplitudes of the desired signal multiplied by the instantaneous amplitudes of the MAI signal, given in sections 3.3 and 3.4, over all possible amplitudes of MAI and over all desired user amplitudes greater than the threshold y/θ.

To compute the probabilities in Eq. 15, we note that

$$P\{\Omega_{n+1} = D|\Omega_n = C, x_{n+1} = j, x_n = i\}$$

which can be written as:

$$\frac{P\{\Omega_{n+1} = D, \Omega_n = C|x_{n+1} = j, x_n = i\}}{P\{\Omega_n = C|x_n = i\}} \tag{18}$$

using Bayes' Theorem. $P\{\Omega_n = C|x_n = i\}$, assuming a stationary channel, is given by Eq. 17.

In order to compute

$$P\{\Omega_{n+1} = D, \Omega_n = C|x_{n+1} = j, x_n = i\} \tag{19}$$

we need to sum all possible instantaneous amplitudes of the desired signal multiplied by the interfering signals for the current timeslot multiplied by the same for the next timeslot. We are only interested in the conditional probability corresponding to *[Good,Good]*, as is evident from Eq. 15. This is given by [14]:

$$P\{\Omega_{n+1} = G, \Omega_n = G | x_{n+1} = j, x_n = i\}$$
$$= \int\limits_0^\infty \int\limits_0^\infty P_{sig}(u_n) \cdot P_{MAI}(y_n < \theta | i) \cdot \left[1 - \int\limits_0^{\frac{y_{n+1}}{\theta}} P_{sig}(u_{n+1}|u_n) du_{n+1} \right] \cdot P_{MAI}(y_{n+1}|j) du_n dy_{n+1}$$

where

$$P_{MAI}(y < \theta | i) = \int\limits_0^\theta P_{MAI}(y|i) dy \tag{20}$$

The following sections provide expressions for $P_{MAI}(u_n|j)$, $P_{sig}(u_n)$, and $P_{sig}(u_{n+1}|u_n)$.

3.3 User signal model

Before deriving an expressions for the user signal at the receiver, we first define the complex fading channel radio signal as done by Turin [21] to be

$$y(t) = c(t)x(t) + n(t) \tag{21}$$

where $x(t)$ is the transmitted signal, $y(t)$ the received signal, and $c(t)$ and $n(t)$ are complex random processes governing the fading and noise distortions respectively. The white noise component $n(t)$ of the signal, will be ignored for our purposes, because mobile radio channels are interference limited and not noise limited [20].

The autocorrelation function of $c(t)$, $R(\tau)$, is given by [19]:

$$R(\tau) = \sigma_u^2 J_0(2\pi f_D |\tau|) \tag{22}$$

Here J_0 is the Bessel function of the first kind, order zero, σ_u^2 is the variance of $u(t)$, the envelope of $c(t)$. The parameter f_D is the channel Doppler bandwidth which is determined primarily by the speed and transmitting frequency of the mobile station. A value $f_D t_s = 0.02$, with t_s the timeslot duration, is considered to represent slow fading, while values of 0.1 and 0.3 represent medium and fast or uncorrelated fading respectively. The results given in Section 6 are for fast fading.

We follow the principles outlined in [21] to determine the relevant PDFs for our signal envelopes, with the exception that our envelope, $\mathbf{u}_k$, for Ricean fading is:

$$\mathbf{u}_k = (w_{u_1}, w_{u_2}, \ldots, w_{u_k}) \tag{23}$$

$$w_{u_i} = \sqrt{A^2 + u_i^2 + 2u_iA\,cos(\theta_i)} \tag{24}$$

where A is signal power of the dominant component. This leads to the following expression, with $k = 1$ in Eq. 8 on page 1810 in [21, Turin] for the Ricean PDF:

$$g(u_1) = \frac{u_1}{\mu} e^{-\frac{u_1^2+A^2}{2\mu}} I_0\left(\frac{Au_1}{\mu}\right) \tag{25}$$

where $\mu = R(0)$ is the local mean scattered signal power [20].

The distribution in Eq. 25 can be expressed in terms of one unknown parameter by defining $K = \frac{A^2}{2\mu}$ where K is known as the *Rice Factor*, and is defined as the ratio between the Line of Sight (LOS) component of the signal and the scattered component [22]. If $K = 0$, the distribution becomes a Rayleigh faded distribution. K is adequate in order to completely specify the Ricean distribution [20, 22].

The general Ricean PDF is therefore given by

$$P_{sig}(u) = \frac{(1+K)}{\bar{p}} e^{-K} u e^{-\frac{1+K}{2\bar{p}}u^2} I_0\left(\sqrt{\frac{2K(1+K)}{\bar{p}}}u\right), u >= 0 \tag{26}$$

$I_0(x)$ is the modified Bessel function of the first kind and order zero [11]. where $\bar{p}$ is the local mean power given by $\bar{p} = \frac{1}{2}A^2 + \mu$.

3.4 Interfering signal model

Due to the imperfect orthogonality of the PN code sets used to multiplex the user signals in DS-CDMA, there is a level of interference that arises as more users access the system. This is called Multiple Access Interference, or MAI. If the MAI is too great, signals become distorted and errors occur. The channel is also a multi-path channel, and this creates yet another source of distortion due to signal reflection.

In order to obtain an expression for the MAI, one needs to take into account a summation of many Ricean random variables, and a closed form expression is not easy to obtain. However, [9] shows that the standard Gaussian approximation is an adequate approximation of the MAI, and can be given by [14]:

$$P_{MAI}(y|j) = \begin{cases} P_{sig}(u), & j = 1 \\ \frac{1}{\sqrt{2\pi\sigma_u^2}} exp\left(\frac{-(y-j\mu_u)^2}{2j\sigma_u^2}\right), & j > 1 \end{cases} \tag{27}$$

which is the PDF of the interference amplitude conditioned on the number j of users present.

An expression for the conditional probability density function of y, $P\{y_n|y_{n-1}\}$ which would consider the correlation between successive samples of the MAI signal is analytically and computationally intractable. The reason is that the contribution from the individual interfering signals and their effect on the desired signal is not known. We therefore make the assumption that the MAI amplitude is uncorrelated which would seem reasonable if the interfering users enter and leave the channel in a manner that the number of transmitting users is practically uncorrelated from time slot to time slot. This assumption is shown to be a valid one in [3].

3.5 Turbo Coding

The Forward-Error Correction (FEC) encoding in the UTRAN takes place in Node B using very powerful Turbo coding. The Downlink Shared Channel (DSCH) in UTRA employs $\frac{1}{3}$-rate coding, which means that a block of data will be roughly three times its original size after it has been encoded. Turbo codes are a simple extension to Recursive Systematic Convolutional (RSC) Codes. The Turbo Coding process involves applying an RSC code to two different versions of the input block of data, one that is in correct order, and another that has been pseudo-randomly permuted. This will produce two blocks of parity data, which are transmitted along with the original data and thereby reducing the effective bandwidth of the DSCH.

The probabilities $h(j)$ and $k(j)$ used in Eq. 14 are considered fixed in any analysis we have seen elsewhere and does not take Turbo coding into account, and thus provides a poor fit to simulated data which accounts for Turbo coding. We therefore condition k on the number j of interfering users and use the formulae presented in [15, Lee] for the upper and lower error bounds of Turbo-coded signals. The error rate of turbo-coded signals lies between two asymptotes, one of which is relevant at a high Signal to Noise ratio (SNR), and one of which is relevant at a low SNR. The formulae for each asymptote are a complex function of the number of interfering users j, amongst others, as well as a transition probability between the two asymptotes given in [15, Lee] and the reader is referred to that work for the details.

Denote the lower bound estimate as $P_e^l(j)$, and the upper bound as $P_e^u(j)$ with probability γ of a transition from the lower to the upper bound. We then write

$$h(j) = \gamma P_e^l(j) + (1-\gamma)P_e^h(j) \tag{28}$$

and assume that $k = 0$ or that no bit errors occur in a good state.

4 Simulation

In this section we provide evidence of the detail in which our discrete event simulator emulated the coding and re-transmission processing in the RCN. This detail is illustrated in Figure 3 which shows all of the queues involved in the acknowledgement and retransmission process. From that figure the RLC entity is responsible for segmenting the PDCP Service Data Unit into RLC Protocol Data Unit's (PDUs), or UTRA transport blocks. The size of the PDU is governed by the choice of transport channel. Our choice of the DSCH means that the RLC protocol must be in Acknowledged Transfer Mode, meaning that the RLC entity must keep track of all RLC PDUs that are transferred to the MAC sub-layer in order to be able to retransmit an error block. The RLC protocol adds a header to each PDU and transfers that to the MAC protocol. The simulator must be able to keep track of all blocks and perform the necessary steps required for retransmissions to occur.

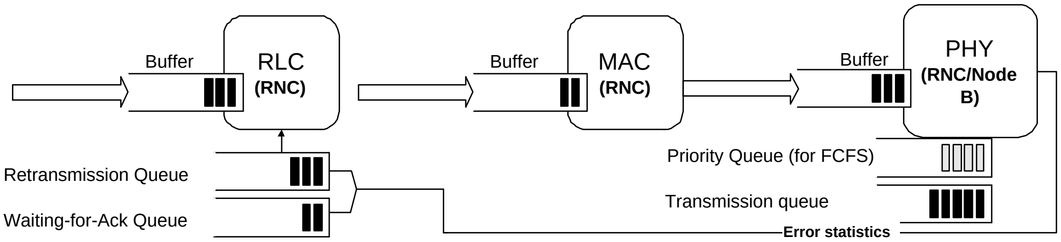

Fig. 3. An overview of the queues involved in the acknowledgement process.

The MAC layer adds a header to each block and then schedules transport blocks onto the DSCH.

The PHY layer performs the majority of the processing required to send the transport blocks over the physical medium. The first process that must occur is the computation of a cyclic redundancy code (CRC) for each transport block which are subsequently Turbo-coded. A Spreading code is then used to encode the blocks which are then ready for transmission with the appropriate channelization code[20].

A transmission occurs every 10 milliseconds. The PHY entity keeps track of all blocks that are waiting for transmission. It also keeps track of the users to whom the packets belong, arranging them in a FCFS queue. The packets belonging to the user at the top of the FCFS queue at each transmission interval are transmitted.

In the simulator transmission is simulated by applying a random number of bit changes to the transmitted blocks. The block is the decoded using a Turbo Decoder. A comparison is made between the original and the decoded block to determine whether the block was correctly received and performance metrics are computed from a output of traced events as is usually done.

5 Model Calibration

Before applying the model we need to determine realistic values for the MAI signal to user signal ratio θ in Eq. 16 and the transition probability γ used in the previous section. Following the example in [14] we matched the values of the error rate $1 - R_l$ from Eq. 8 for a range of values of θ with results from a simulation. The results are shown in Figure 4. Note that the simulated values are all zero below $j = 10$ since errors are eliminated by the Turbo coding process. From the figure it is clear that for small $j < 21$ a value of $\theta = 35$ provides the best fit. Since the error rate approaches 1 for higher values of j we decided to use $\theta = 35$ in the remainder of our analysis.

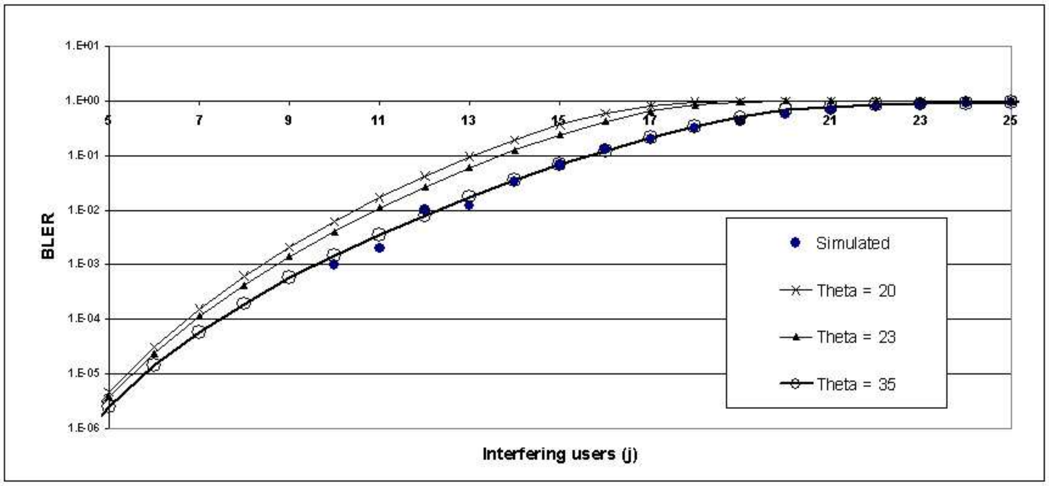

Fig. 4. Analytic error rate $1 - R_l$ (Eq. 8) and simulated values for various values of θ

The value for γ used in our analysis in Eq. 28, we derived by choosing a threshold number of users. When j is larger or equal to this threshold, the performance matches the upper bound $P_e^u(j)$ and tends towards this upper bound for j less than the threshold. We then derived the empirical expression

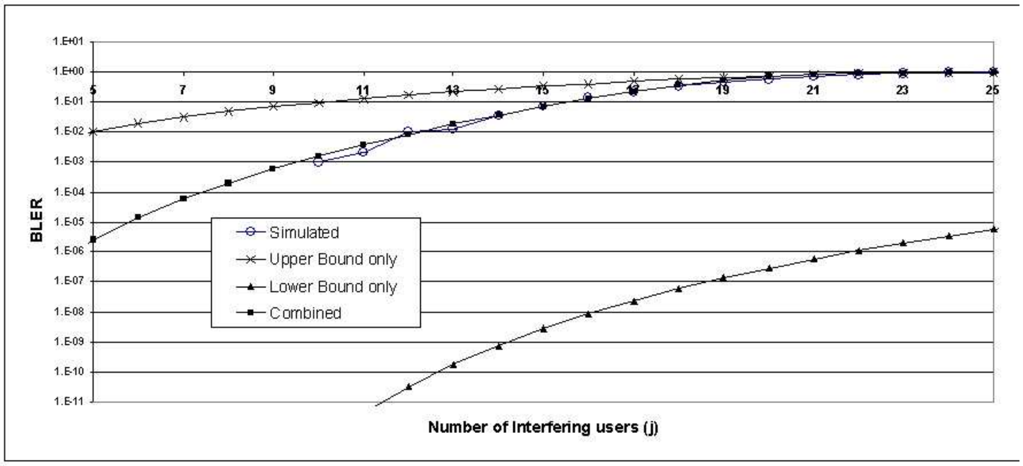

Fig. 5. Analytic approximation (combined) of the block error rate BLER or $h(j)$ for j interfering users

$$\gamma(j) = \begin{cases} 1 - \left(\frac{j}{\kappa}\right)^6 & j < 20 \\ 0 & j \geq \kappa \end{cases} \tag{29}$$

in an attempt to approximated this error behavior as a function of $h(j), j = 1, \ldots$ For a threshold value of 20 users the function given in Eq. 29 we believe an adequate approximation as can be seen from Figure 5.

6 Model Validation and Results

Ideally one should be able to measure the delay times in a real mobile network to verify the analytical model described in this paper. This is near impossible and in order to validate the model we therefore compared the analytic results with those from the simulator described in Section 4. Figure 6 shows the overall model delay for increasing packet arrival rates for various numbers of packet arrival streams and CDMA users. From that figure it is clear that, except at overload conditions when the packet arrival rates are larger than 150 per second, the analytic results are an excellent approximation to the simulation results. We therefore believe that the analytic model is an adequate representation of the system we modeled.

The results already shown gave us sufficient confidence in the model to perform various experiments such as that illustrated in Figure 7 of the throughput of

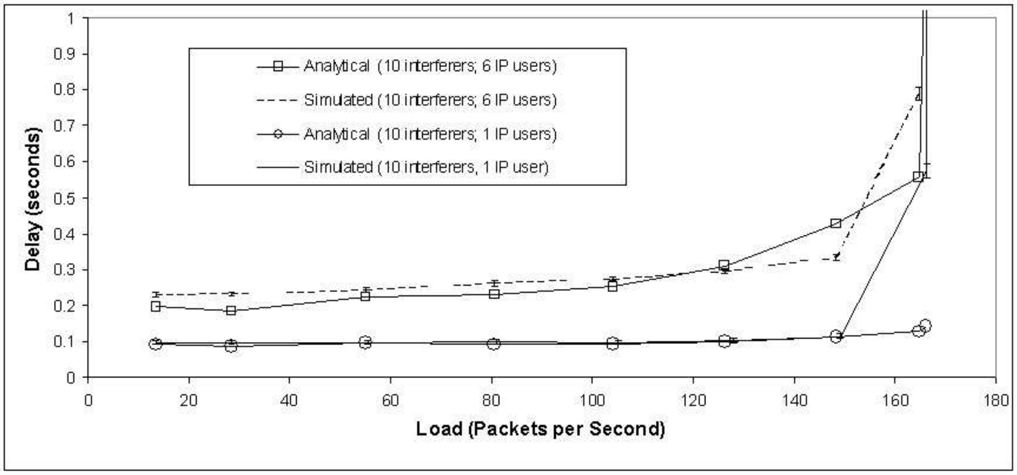

Fig. 6. Simulated and analytical mean delay time versus packet arrival rate

IP packets as a function of the number of interfering users. Note that the delay time shown is on a logarithmic scale. The results would seem to indicate that at high arrival rates the performance of the mobile link will become unacceptable for very few concurrent users, thus indicating that UMTS may not be a sensible alternative technology to WiFi in an environment where both could be used.

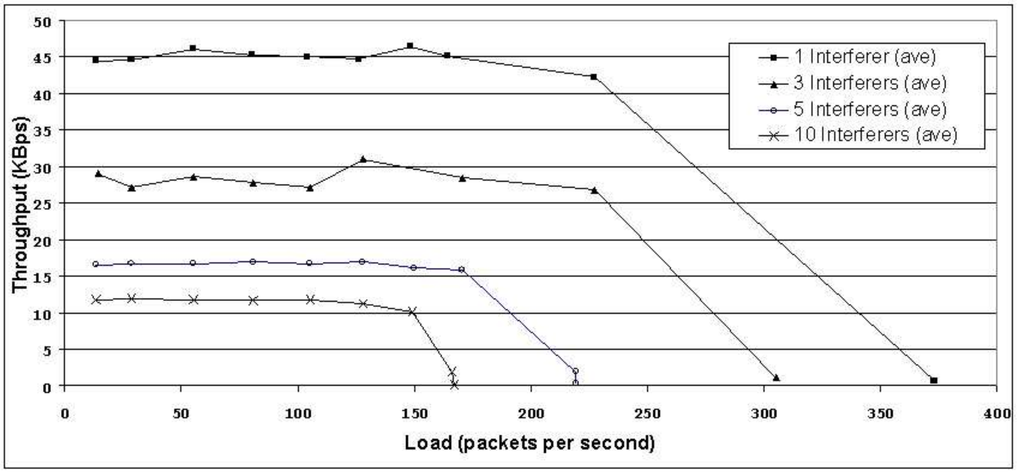

Fig. 7. Analytic packet delay versus number of interfering users for various packet arrival rates

7 Conclusion

We have presented an analytic model for the down link delay of IP traffic between the Mobile Gateway Server and the End User in a UMTS mobile network. The model uses a Hidden Markov Model to analyze the physical channel characteristics, and a BMAP/D/1 queue to model the path from IP network to the Radio Network Controller. The physical channel model takes into account Ricean fading and multiple access interference. It also takes into account the net effect of Turbo coding. The analytical model was calibrated using a discrete event simulator and with the parameters so determined, subsequent results were verified with the same discrete event simulator. The model is complex and relies on the results of several others as indicated in the paper. We believe that the effort was nevertheless worth it and that the model is a useful tool to quickly confirm or otherwise intuitive network behavior.

References

[1] A. Klemm, C. Lindemann and M. Lohmann. Traffic Modeling and Characterization for UMTS Networks. In *Globecom 2001 Internet Performance Symposium*, pages 1741 – 1746, 2001.

[2] A. Klemm, C. Lindemann and M. Lohmann. Modeling IP Traffic Using the Batch Markovian Arrival Process. *Performance Evaluation* , 54:149–173, 2003.

[3] J Cheng and N Beaulieu. Accurate DS-CDMA bit-error probability calculation in Rayleigh fading. *IEEE Transactions on Communications*, pages 3 – 15, January 2002.

[4] David M. Lucantoni. New results on the single server queue with a batch Markovian arrival process. *Stochastic Models*, 7(1):1–46, 1991.

[5] David M. Lucantoni. The BMAP/G/1 queue: A tutorial. *Models and techniques for Performance Evaluation of Computer and Communications Systems*, pages 330–358, 1993.

[6] E.O. Elliot. Estimates of error rates for codes on burst-noise channels.

[7] A. Fiorini. Uplink user bitrate adaptation for packet data transmissions in wcdma, 2000.

[8] E Geraniotis. Direct-sequence spread-spectrum multiple-access communications over nonselective and frequency-selective Rician fading channels. *IEEE Transactions on Communications*, pages 756 – 764, August 1986.

[9] E Geraniotis and M Pursley. Performance of noncoherent direct-sequence spread-spectrum communications over specular multipath fading channels. *IEEE Transactions on Communications*, pages 219 – 226, March 1986.

[10] H Bourlard and N Morgan. Hybrid HMM/ANN Systems for Speech

Recognition: Overview and New Research Directions. *Adaptive Processing of Sequences and Data Structures, Volume 1387 of Lecture Notes in Artificial Intelligence*, pages 389–417, 1998.

[11] Simon Haykin. *Communications Systems.* John Wiley and Sons, Inc., 2001.

[12] J Abate and W Whitt. The Fourier-Series Method for Inverting Transforms of Probability Ditributions. *Queueing Systems*, 1991.

[13] D. Johnson. Validation of wireless and mobile network models and simulation, 1999.

[14] Garth Judge and Fambirai Takawira. A simple hidden Markov model for a CDMA channel with correlated Rayleigh fading. *The Transactions of the SAIEE*, March:17–26, 2002.

[15] J. W. Lee and R. E. Blahut. Bit error rate estimate of finite length Turbo codes. In *IEEE 2003 International Conference on Communications (ICC 2003), Anchorage, AK*, May 2003.

[16] C. Lindemann and M. Lohmann. IP2BMAP software package. Available online: www.ip2bmap.de.

[17] Laurence B. Milstein Michele Zorzi, Rameh R. Rao. Error statistics in data transmission over fading channels. *IEEE Transactions on Communications*, 46:1468–1477, 1998.

[18] Erwin Mondre. Complex and envelope covariance for Rician fading communication channels. *IEEE Transactions on Communications*, pages 80 – 84, February 1971.

[19] Cecilio Pimentel and Ian F. Blake. Modelling burst channels using partitioned Fritchman's Markov models. *IEEE Transactions on Vehicular Technology*, 47:885–899, 1998.

[20] T.S. Rappaport. *Wireless Communications: Principles and Practices.* Prentice Hall, 1996.

[21] William Turin and Robert van Nobelen. Hidden Markov modeling of flat fading channels. *IEEE Journal on Selected Areas in Communications*, 16:1809–1817, 1998.

[22] Michele Zorzi. Capture probabilities in random access mobile communications in the presence of Ricean fading. *IEEE Transactions on Vehicular Technology*, Feb, 1997.

A Pieter Kritzinger

Pieter obtained an M.Sc. in Electrical Engineering from the University of the Witwatersrand in Johannesburg South Africa and his PhD in Computer Science from the University of Waterloo, Canada, where he became Assistant Professor for 2 years.

This was followed by 2 years of teaching at Imperial College, University of

London before he returned to South Africa to become a senior lecturer at Stellenbosch University. He was invited to join the Univresity of Cape Town (UCT) as a full professor and was appointed without advertisement in July 1985; something which is very rarely done. Pieter was Head of the Department of Computer Science for the period January 1989 to December 1996 when his term ended.

His research interests are in the role of Formal Description Techniques in developing reliable software and stochastic models of concurrent communicating systems.

During his career in South Africa, he has spent some 3 years in all on study leave at the Abteilung für Informatik at the Universität Dortmund (most recently in 1992) IBM's Zürich Research Laboratory (during 1984 and again in 2004), the Universität Erlangen-N"urnberg (most recently for two months during 2002) and the Institut National des Télécommunications in Evry, France. Pieter was elected a Senior Member of the American Institute for Electronic and Electrical Engineers in September 1998. He is a member of IFIP Working Groups 6.1, 6.3 and 7.3.

Fig. A.1. Pieter Kritzinger

B Jesse Landman

Jesse Landman obtained his B.Sc. in Computer Science and Applied Mathematics from the University of Cape Town (UCT) in 2001. He completed his Honors degree studies with the Data Networks Architecture Group in the Department of Computer science, concentrating on Formal Description Techniques and their application to real-time voice networks. He began his M.Sc. research in 2003 and graduated in 2005 with the dissertation "Models of IP Traffic over UMTS networks". Jesse is currently working as a software developer in Cape Town, South Africa.

Fig. B.1. Jesse Landman