QUEUEING MODEL OF THE AAL2 MULTIPLEXER IN UTRAN

Gábor Horváth
Department of Telecommunication
Budapest University of Technology and Economics
1521, Budapest Pf. 91., Hungary
E-mail: ghorvath@hit.bme.hu

Csaba Vulkán
Nokia Research Center
1092 Budapest, Köztelek u. 6., Hungary
E-mail: csaba.vulkan@nokia.com

KEYWORDS

UMTS, ATM, AAL2 multiplexer, Timer_CU, BMAP/D/1 queue, matrix analytic solution

ABSTRACT

This paper presents a queueing model for the performance evaluation of the AAL2 multiplexer in UTRAN (UMTS Terrestrial Radio Access Network). Based on the model analytical expressions are provided for the distribution of the waiting time and for the multiplexing efficiency. The performance of the AAL2 multiplexer is studied, especially the impact of the Timer_CU on the QoS parameters. The analytical results are verified with simulations.

1 INTRODUCTION

As UMTS (Universal Mobile Telecommunication System) is currently under deployment, the system's performance improvement and optimization is of special interest. In addition to introducing improvements that aim to increase the system capacity and the users' peak throughput for data services, the optimal usage of transport resources must be also achieved. The WCDMA radio control functions are imposing strict delay requirements [3] both for RT (real-time) and NRT (non real-time) services over the transport network between the RNC (Radio Network Controller) and Node B (Base Station). These requirements should be guaranteed with the maximal utilization of the transport capacity that is a limited resource especially on the last mile links (typically 1xE1 or 2xE1).

ATM/AAL2 (Asynchronous Transfer Mode/ATM Adaptation Layer Type 2 [1]) has been selected as transport network layer for the UMTS Iub interface that connects the RNC and the Node B, because it is able to multiplex several voice and data connections into one VCC (Virtual Circuit Connection); improving in this way the utilization of the transport network. The delay on the transport network consists of the AAL2 multiplexing delay (the delay on the AAL2 layer) and the delay on the ATM layer. Assuming CBR (Constant Bit-Rate) VCCs we can consider that the former is the dominating delay component on the transport network. If no AAL2 switching is implemented in the transport network, the delay on the AAL2 multiplexer should be analyzed at the RNC for the downlink traffic. Since the traffic on the AAL2/ATM transport network is delay sensitive, the buffering delay caused by the AAL2 multiplexer is allowed to exceed a given value (maximum allowed delay) only with low probability. The maximum allowed delay and the probability of exceeding it are the basis of the ATM VCC and link capacity dimensioning, and also the reference value when the quality of the transport network is evaluated. Additionally, the multiplexing efficiency is evaluated with a measure called packing density. The efficiency is the highest when the payload of the ATM cells is fully utilized (does not contain padding). In order to increase the efficiency, a timer (called Timer_CU) is introduced at the AAL2 multiplexer.

The performance of AAL2 multiplexing with different traffic types and multiplexer settings was examined analytically in [12], [6], [13] and [10], by simulation [8], [9], [11], [15] or by both analytical models and simulation [7].

While some papers are focusing on the buffer requirement and delay issues [15] others are investigating the differences between Assembly Before Transmission (ABT) and Combined Assembly and Transmission (CAT) multiplexing method and studying the effect of changing the Timer_CU value [9], [7] or the various scheduling algorithms that can be used [8]. Analytical model of the ABT architecture with Timer_CU assuming Poisson arrivals is described in [12]. Another analytical model is provided by [13] with Poisson arrivals and no Timer_CU. Batch Bernoulli traffic model, no Timer_CU and frame sizes multiple of the ATM cell payload is assumed in [10] and [6]. Some papers are comparing the performance of different adaptation layers (AAL1, AAL2, AAL5) in case of transporting voice over ATM network [11], [15]. According to our knowledge, there are no research results published on the performance analysis of CAT AAL2 multiplexers with less restrictive traffic models.

In this paper we introduce and analyze a CAT AAL2 multiplexer with Timer_CU. The traffic model is a batch markovian arrival process (BMAP). We show that the embedded process at departures is a Markov chain of M/G/1 type, which can be efficiently analyzed by matrix geometric methods. We provide methods to compute the two most crucial performance measures of the AAL2 multiplexer, namely the distribution of the waiting time and the multiplexing efficiency.

The rest of the paper is organized as follows. Section II discusses the AAL2 multiplexer. Section III provides a detailed overview of the analysis of the BMAP/D/1-Timer queuing system. Numerical results are summarized in Section IV. Section V concludes the paper.

2 THE AAL2 MULTIPLEXER

The user plan protocol stack of RNC - Node B interface (Iub) consists of Radio Network and Transport Network Layers [1]. FP (Frame Protocol) creates frames out of the user traffic mapped into Dedicated Channels (DCH) and sends them through the transport bearers i.e. AAL2 connections (Figure 2) at each Transmission Time Interval (TTI).

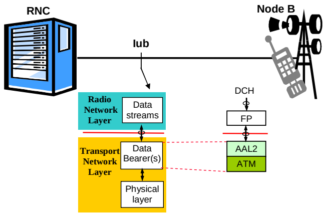

Figure 1: Protocol stack of the Iub interface

The AAL2 layer is multiplexing up to 248 connections into one ATM VCC. The incoming FP frames (AAL2-SDUs) are segmented by the Service Specific Convergence Sublayer (SSCS) into maximum 45 byte segments. The Common Part Sublayer (CPS) encapsulates these segments by adding a 3 bytes header. The encapsulated segments are called CPS-Packets and their size is at maximum 48 bytes. The AAL2 multiplexer puts the CPS-Packets into CPS-PDUs (with 1 byte header referred to as STF), which are in fact the payload of the ATM cells and their maximum size is 48 bytes. This means that a full sized CPS-Packet can only be transported in two ATM cells. In order to increase the multiplexing gain a timer (Timer_ CU) is initialized whenever the CPS-PDU is smaller than 48 bytes i.e. the CPS-Packets under assembly are not filling an ATM cell.

The ATM cell is padded and sent when Timer_CU expires and there is no new CPS-Packet arriving to the multiplexer. Setting larger value for the Timer_CU results in larger delay and higher multiplexing gain, i.e., the load of ATM links is reduced.

This paper focuses on the analysis of the AAL2 multiplexing delay and multiplexing efficiency as the function of the available bandwidth and on the impact of the Timer CU in case of CBR VCC.

3 ANALYSIS OF THE AAL2 MULTIPLEXER MODEL

Based on the description presented in the previous section, we have created a queueing model for the AAL2 multiplexer.

In this paper we assume that the inter-arrival time and size variation of the CPS-Packets is given by a BMAP traffic descriptor. The CPS-Packets are stored in the multiplexing buffer. In our model, the buffer size is measured in bytes, thus if we say that the queue length is k, it means that k bytes are waiting in the buffer.

The server transmits the CPS-Packets multiplexed into CPS-PDUs. The size of the CPS-PDU payload L is constant ($L = 47$ bytes). As soon as a CPS-PDU is assembled, it is encapsulated into an ATM cell, and transmitted (see Figure 2). Since the transmission happens on a constant bit rate channel (CBR VCC), the service time is deterministic (Δ).

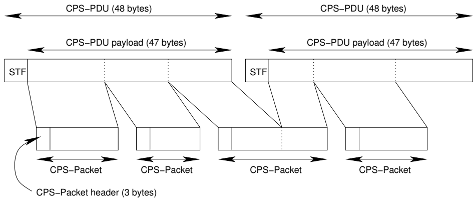

Figure 2: Multiplexing CPS-Packets into ATM cells

To model the buffer of the multiplexer, we apply the BMAP/D/1-Timer queueing model. This system is introduced in [5]. In that paper the derivation and a numerical model for the waiting time distribution is provided. In the following sections we briefly summarize the results of that paper, and extend it to compute the multiplexing efficiency.

3.1 Traffic Model

As mentioned above, the traffic arriving to the multiplexer is described by a continuous-time batch markovian arrival process with m phases. Its $m \times m$ generator is denoted by D. The arrival process itself is characterized by a set of matrices D_i ($\sum_{i=0}^{K} D_i = D$), where $[D_k]_{i,j}$ corresponds to the arrival of a k bytes long CPS-Packet followed by a state transition from i to j.

The probability that there are n arrivals in time t with the arrival process being in state i at the beginning and in state j at time t is denoted by $[P(k,t)]_{i,j}$.

The BMAP is a strong modeling tool for traffic description in markovian analysis. It can be created based on measured or approximated real traffic behavior (see [4]). Although deterministic traffic can not be captured accurately by BMAPs, former research ([14]) pointed out that the superposition of deterministic sources behaves as a Poisson process (the more on-off modulated deterministic traffics are superposed, the closer is the aggregate to the Poisson process).

The BMAP characterization of the traffic in UTRAN is out of scope of this paper, it is the subject of further research.

3.2 Behavior of the Timer

The timer ensures that when data reaches the head of the queue (positions 1 to L, thus it will be included in the next CPS-PDU), it will be served in time T, even if there is not enough data to fill a complete CPS-PDU when the timer expires. In that case, the server pads up the a partially filled CPS-PDU, and transmits it with the same service time (Δ).

To give a better description of the service mechanism, we summarize the behavior in the following three points. These three points will be referred many times in the sequel, and will be used as three cases requiring different treatments during the analysis:

P1. While there are $\geq L$ bytes in the queue, the timer is not started. The server takes the first L bytes out of the queue, compiles the CPS-PDU, and starts the transmission.

P2. If the queue size is $< L$ but > 0 when the transmission begins (after taking out the ones whose transmission has started), the timer is started. When the service is ready and the queue is still $< L$, no new service begins until the timer elapses, or the sufficient amount of data arrives.

P3. If an arrival happens when the queue is empty, and the batch size of the arrival is below L, the timer is started immediately. The first service can start when the the queue size exceeds L, or when the timer elapses. Of course the service can not start until the CPS-PDU under transmission does not leave the multiplexer.

3.3 Queue Length Distribution at Packet Departures

The state of the multiplexer buffer can be characterized by a discrete time Markov chain at CPS-PDU assembly instants. These CPS-PDU assembly instants can be translated as data departure events from the multiplexer buffer. Embedding at departures usually leads to a so called M/G/1 type structure, as it does in our case, too. The transition probability matrix builds up as follows:

$$\boldsymbol{\mathcal{X}} = \begin{bmatrix} \boxed{\qquad\quad \boldsymbol{\mathcal{B}} \qquad\quad} \cdots \\ \boxed{\qquad\quad \boldsymbol{\mathcal{A}} \qquad\quad} \cdots \\ \qquad \boxed{\qquad \boldsymbol{\mathcal{A}} \qquad} \cdots \\ \qquad\qquad \boxed{\quad \boldsymbol{\mathcal{A}} \quad} \cdots \\ \qquad\qquad\qquad \boxed{\ \boldsymbol{\mathcal{A}}\ } \cdots \\ \qquad\qquad\qquad\qquad \ddots \quad \vdots \end{bmatrix}. \qquad (1)$$

Usually M/G/1 type matrices are defined by their quadratic matrix blocks. Now we define the matrix by its block rows, because it simplifies the definition. The states inside the $m \times m$ blocks are reflecting the state of the arrival process; a transition from block i to block j means that the queue size changed from i to j since the last departure instant.

Matrix row $\boldsymbol{\mathcal{A}}$ corresponds to case P1 of Section 3.2. In this case the inter departure time is exactly Δ, since the timer does not play a role. At the next embedded point the server will decrease the queue by L, so the queue size change equals to the arrivals during Δ minus L. Thus, $\boldsymbol{\mathcal{A}}$ is defined by:

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} P(0,\Delta) & P(1,\Delta) & P(2,\Delta) & \dots \\ & P(0,\Delta) & P(1,\Delta) & \dots \\ & \ddots & & \vdots \\ & & P(0,\Delta) & \dots \end{bmatrix}$$

(This matrix has $L \times m$ rows).

The definition of $\boldsymbol{\mathcal{B}}$ is more complex due to the effect of the timer. We further divide $\boldsymbol{\mathcal{B}}$, to distinguish between the completely idle (P3) and not completely idle (P2) cases:

$$\boldsymbol{\mathcal{B}} = \left[\begin{array}{c} \hat{\boldsymbol{\mathcal{B}}} \\ \hline \\ \tilde{\boldsymbol{\mathcal{B}}} \\ \\ \end{array}\right]. \qquad (2)$$

The first block row (with height m) describes the transitions from the idle buffer (these matrices are denoted by $\hat{\boldsymbol{\mathcal{B}}}$, and correspond to P3 in Section 3.2), the other rows describe transitions from buffer levels 1 – L-1 (P2 in Section 3.2).

In both cases the timer is started, and the next transition may begin no sooner than Δ. The evolution of the queue size between levels 1 and L is important to capture the effect of the timer. Therefore we define the continuous time Markov chain generator $\boldsymbol{Q}$, that follows the queue size increase process between 1 and L:

$$\boldsymbol{Q} = \begin{bmatrix} D_0 & D_1 & \dots & D_{L-2} \\ & D_0 & \dots & D_{L-3} \\ & & \ddots & \vdots \\ & & & D_0 \end{bmatrix}, \qquad (3)$$

and $\boldsymbol{Z}(t)$ which is the transition probability matrix of the buffer size increase process during time t, thus, $\boldsymbol{Z}(t) = e^{\boldsymbol{Q}t}$.

The behavior of the timer is described by $\boldsymbol{\Pi}(t)$. $[\boldsymbol{\Pi}(t)]_{i,j}$ is the probability that starting with i as initial state (with buffer length between 1 and $L-1$) the state of system will be j just after the start of the next service. The next service can start when the necessary number of bytes have arrived until time t (first term in eq. (4)), or at time t the buffer content is served even if a full CPS-PDU can not be created (second term of eq. (4)). This probability matrix is computed by:

$$\mathbf{\Pi}(t) =$$

$$\int_0^t e^{\boldsymbol{Q}\tau} d\tau \cdot \begin{bmatrix} D_{L-1} & \dots & D_K & & & \\ D_{L-2} & \dots & D_{K-1} & D_K & & \\ \vdots & \ddots & \dots & \dots & D_K & \\ D_1 & \dots & \dots & \dots & \dots & D_K \end{bmatrix} \quad (4)$$

$$+ e^{\boldsymbol{Q}t} \cdot \begin{bmatrix} I_{m\times m} & 0 & 0 & \dots \\ I_{m\times m} & 0 & 0 & \dots \\ \vdots & \ddots & & \\ I_{m\times m} & 0 & 0 & \dots \end{bmatrix}.$$

If the buffer size is between 1 and $L-1$ ($\tilde{\boldsymbol{\mathcal{B}}}$), the number of arrivals during the Δ interval (during which the server is busy) has to be investigated. If the arriving CPS-Packets increase the buffer above L, the assembly and transmission of a new CPS-PDU starts just after finishing the previous one (this events are expressed by the first matrix term of eq. (5)). If the buffer level is still below L (second term of eq. (5), an additional delay follows, with a maximal length of $(T-\Delta)^+$, since the timer started at the moment when the buffer decreased below L. Thus:

$$\tilde{\boldsymbol{\mathcal{B}}} = \begin{bmatrix} P(L-1,\Delta) & P(L,\Delta) & \dots \\ P(L-2,\Delta) & P(L-1,\Delta) & \dots \\ \vdots & \vdots & \dots \\ P(1,\Delta) & P(2,\Delta) & \dots \end{bmatrix} + \quad (5)$$

$$+ \boldsymbol{Z}(\Delta) \cdot \mathbf{\Pi}((T-\Delta)^+).$$

If the buffer empties at the beginning of the service of a packet ($\hat{\boldsymbol{\mathcal{B}}}$), we have two cases. First, it is possible that there are no arrivals during the server occupancy time (Δ). In this case the next arrival can bring the queue above L, immediately causing a start of the transmission of a CPS-PDU (first term of eq. (6)); or the queue remains below L, and a waiting period for more CPS-Packets follows with a maximum length given by T, since the arrival into the empty queue initiated the timer (second term of eq. (6)). The second case is when there were arrivals during the server occupancy time. In this case the arrivals can bring the queue above L, and the service of a new CPS-PDU starts just after the end of service of the previous one (third term of eq. (6)). The arrival can leave the system below L, and a waiting period is started with a maximal length of $(T-(\Delta-\tau))^+$, since the timer started at the first arrival time (τ) and from the timer $\Delta-\tau$ time already expired when the server becomes empty (fourth term of eq. (6)). Thus we have:

$$\begin{aligned} \hat{\boldsymbol{\mathcal{B}}} &= P(0,\Delta)(-D_0)^{-1} \; D_L \quad D_{L+1} \quad \dots \quad D_K \; + \\ &+ P(0,\Delta)(-D_0)^{-1} \; D_1 \quad D_2 \quad \dots \quad D_{L-1} \; \cdot \mathbf{\Pi}(T) + \\ &+ \; P(L,\Delta) \quad P(L+1,\Delta) \quad \dots \; + \\ &+ \int_0^{\Delta} e^{D_0\tau} \cdot \; D_1 \quad D_2 \quad \dots \quad D_{L-1} \; \cdot \boldsymbol{Z}(\Delta-\tau) \cdot \\ &\quad \cdot \mathbf{\Pi}((T-\Delta+\tau)^+) d\tau. \end{aligned} \quad (6)$$

The steady state distribution of the embedded Markov chain (1) is partitioned the following way:

$$\boldsymbol{x} = \Big[\underbrace{p_0 \quad p_1 \quad \dots \quad p_{L-1}}_{\boldsymbol{x}_0} \; \underbrace{p_L \quad p_{L+1} \quad \dots \quad p_{2L-1}}_{\boldsymbol{x}_1} \; \dots \Big],$$

where p_i is a vector of size m. The steady state probability vector can be efficiently obtained by a matrix analytic method summarized in [5].

3.4 Waiting Time Distribution

The waiting time distribution $P(W > w)$ is the probability that the waiting time of an arriving CPS-Packet (measured from its arrival to its departure) exceeds a given threshold w. It can be easily computed if some parameters are kept fixed.

These parameters are: the length of the buffer at the arrival (k), the remaining server occupation time (t_1), and the maximal departure time measured from the point when the the buffer descends below level L (t_2). If we know the particular values of these parameters, the waiting time distribution $P_W(k,t_1,t_2)$ can be computed by the following way:

$$P_W(k,t_1,t_2) =$$

$$\begin{cases} h & \text{if } w < t_1 + \frac{k}{L} \Delta, \\ 0 & \text{if } k<L, w\geq t_1+t_2+ \frac{k}{L} \Delta, \\ & \text{if } k\geq L, w\geq t_1+(T-\Delta)^+ + \frac{k}{L} \Delta, \\ \left[e^{\boldsymbol{Q}(W^*-\Delta)} h\right]_{\{k/L\}} & \text{otherwise,} \end{cases}$$

where h is a vector of ones with size m.

The first item corresponds to the case when the server occupancy time plus the service time of the CPS-Packets in the queue exceeds the waiting time requirement. In this case $P(W > w)$ equals to one.

The second item covers the case when the waiting time requirement is surely satisfied. This happens if the waiting time requirement is larger than the server occupancy time, plus the service time of the packets in the queue, plus the maximal possible delay caused by the timer. The latter quantity is t_2 if the buffer size is less than L (this is the definition of t_2). It is $(T-\Delta)^+$ if $k \geq L$, because the server will be occupied when the last CPS packet gets below L in the buffer.

The third item means that the waiting time exceeds w if the L long block in the queue, in which k belongs to, is still not filled up until $w-\Delta$. $[e^{\boldsymbol{Q}t}]_i$ is the probability that the arrival process did not generate enough arrivals to leave this block until time t, if there were i bytes in the block at the beginning. $\{k/L\}$ (where $\{\}$ denotes the remainder of the division) is the buffer position inside the L long block after the arrival.

To obtain the waiting time distribution, we have to multiply $P_W(k,t_1,t_2)$ by the probability of the given k, t_1 and t_2 parameters. For all the details and an efficient numerical method see [5].

3.5 Multiplexing efficiency

The multiplexing efficiency is characterized by $\eta \in (1/L, 1]$. It equals to 1, if the payloads of all departing ATM cells are fully utilized, and η is small if the departing ATM cells are containing only few bytes as useful payload. The multiplexing efficiency is calculated from of the departure rate μ and the arrival rate λ as $\eta = \lambda/(L\mu)$.

In the rest of this section we compute the mean departure time $E(D)$ that is the inverse of μ. Again three cases are distinguished, according to P1, P2 and P3. To model the effect of the timer to the mean departure time, we introduce the $m \times L$–1 sized vector $E(W_T(t))$. The kth m sized block in $E(W_T(t))$ is the mean waiting time till the buffer size increases above L, or it is t if it is still less than L at t. $E(W_T(t))$ is computed by:

$$E(W_T(t))=\int_0^t \tau e^{\boldsymbol{Q}\tau}d\tau \begin{bmatrix}\sum_{k=L-1}^{\infty} D_k h\\ \sum_{k=L-2}^{\infty} D_k h\\ \vdots \\ \sum_{k=1}^{\infty} D_k h\end{bmatrix} + te^{\boldsymbol{Q}t}h_{m\times L-2}$$

P1. $E(D_1)$ is the mean departure time under the condition that at the last departure the queue size was not less than L. The departure time is Δ, since there are enough bytes to assemble a new CPS-PDU just after finishing the last one:

$$E(D_1) = \Delta h_m,$$

where h_m means a vector of ones with size m.

P2. $E(D_2)$ is the mean departure time under the condition that at the last departure the queue size was between 1 to $L-1$. The server is occupied for time Δ (first term of eq. (7)). If the queue size remains $< L$ when the server becomes idle, the departure time increases due to the timer (second term of eq. (7)):

$$E(D_2) = \Delta h_{m\times L-2} + Z(\Delta)E(W_T((T-\Delta)^+)). \quad (7)$$

P3. $E(D_3)$ is the mean departure time under the condition that the last departure has left the queue empty. The server is busy for time Δ (first term of eq. (8)). If the first arrival arrives in $(0, \Delta)$, and the queue is still below L at Δ, the departure time is increased by and additional delay caused by the timer. This is reflected by the second term of eq. (8). If the first arrival arrives after Δ, and its size is less than L, the timer is started. In this case the departure time is increased by the time of the first arrival (its mean value is $(-D_0)^{-1}$, and by the additional delay of the timer (third term of eq. (8)):

$$\begin{aligned}E(D_3) &= \Delta h_m + \\ &+ \int_0^\Delta e^{D_0 t}[D_1 \dots D_{L-1}]Z(\Delta - t)E(W_T((T-\Delta+t)^+)dt + \\ &+ e^{D_0\Delta}\ (-D_0)^{-1}h_m + (-D_0)^{-1}[D_1 \dots D_{L-1}]E(W_T(T))\ .\end{aligned} \quad (8)$$

Using these conditional mean departure times, the departure intensity is expressed by:

$$\mu = \frac{1}{p_0E(D_3)+[p_1 \dots p_{L-1}]E(D_2)+\sum_{k=L}^{\infty} p_k E(D_1)}$$

4 NUMERICAL RESULTS

We have implemented the computation method in MATLAB, and also wrote a simulation tool in Omnet++ ([2]) to check the correctness of both the expressions and the MATLAB implementation. Most of the figures in this section are showing both the MATLAB (with lines) and the simulation results (indicated with points).

In the numerical examples, the BMAP of the packet arrivals is a "naive" model of N AMR12.2 voice channels ($N = 5$). These voice channels have an "on-off" behavior, with exponentially distributed "on" and "off" durations. The mean "on" period is 1.5 sec and the mean "off" period is 1.0 sec. During the "on" period the inter arrival times are exponentially distributed with a mean of 20 milliseconds. A (deterministic) 37 byte long data frame (including 5 byte FP header) is generated at each arrival. The 37 bytes long data frame together with the 3 bytes long CPS-Packet header gives 40 bytes long CPS-Packets.

To decrease the computation time, we compress all size-related quantities by 6. Thus, the size of the CPS-PDU is $L = 47/6 \approx 8$, and the size of the CPS-Packets is $40/6 \approx 7$ in the following examples.

In our first example the service time is varying between 0 and 5 ms, and we examine the probability of exceeding the $w = 5$ ms waiting time. The results are depicted in Figure 3.

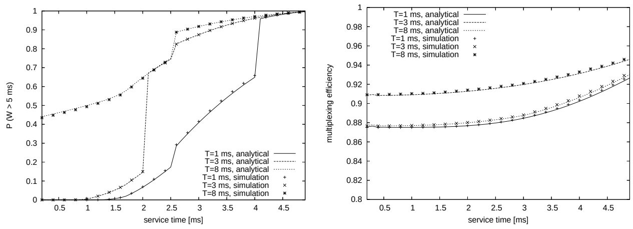

Figure 3: $P(W > 5$ ms) and the multiplexing efficiency vs. the service time

There are probability masses at $\Delta = w/2 = 2.5$ and at $\Delta = w - T$. For the exhaustive explanation of this phenomenon see [5]. From an engineering aspect, Figure 3 reflects also that if $T > w$ holds, the waiting time requirement can not be satisfied even by infinite link capacity (see the curve of $T = 8$ ms), it does not decrease to 0 as the service time tends to 0 (that is, the link capacity tends to infinity). Figure 3 shows that the multiplexing efficiency is better with higher timer values.

In the last example we investigate the effect of increasing the input traffic of the queue by increasing parameter N. Figure 4 shows that if $T + \Delta < w$ holds, the waiting time increases, because the queue size increases. But, if $T + \Delta > w$ (as at $T = 8$), the effect is the opposite, since with increasing traffic the probability that the packet is sent due to the timer – thus, the probability of exceeding the 5 ms requirement – decreases.

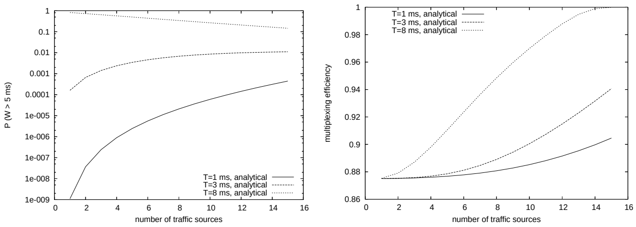

Figure 4: $P(W > 5 \text{ ms})$ and the multiplexing efficiency vs. number of traffic sources

In all the examples the computation of one point of the waiting time distribution took few seconds on a Pentium4 2.4 GHz machine with our MATLAB implementation, while the simulation gave acceptable results only in few minutes. Of course, with a C implementation the speed of the analytical algorithm can be increased substantially.

5 CONCLUSION

This paper introduces a queuing model of the AAL2 multiplexer in UTRAN and its performance analysis.

The introduced analytical model reflects practically important properties of real systems. This way the numerical analysis of the model allows to investigate the effect of such crucial parameters like the Timer_CU. The analysis quantified the intuitive expectations, namely that the higher is the Timer_CU the better is the packing efficiency, but the higher is the CPS-Packet delay at the same time. Indeed, the effect of higher CPS-Packet delay is so significant that higher link capacity cannot compensate it as the Timer_CU tends to the maximal allowed delay.

The presented analytical results are verified by discrete event simulation, and the results show a good accuracy of the introduced analysis procedure.

References

[1] ITU-T 363.2; B-ISDN ATM adaptation layer specification: Type 2 AAL.

[2] OMNeT++ Discrete Event Simulation System, http://www.omnetpp.org.

[3] Technical Specification Group (TSG) RAN; delay budget within the access stratum TR 25.932 V1.0.0 (2000-05).

[4] A. Horváth, G. I. Rózsa, and M. Telek. A MAP fitting method to approximate real traffic behaviour. In *8th IFIP Workshop on Performance Modelling and Evaluation of ATM & IP Networks*, pages 32/1–12, Ilkley, England, July 2000.

[5] G. Horváth and M. Telek. Analysis of a BMAP/D/1-Timer multiplexer. In *First International Workshop on Practical Applications of Stochastic Modelling, PASM'04*, pages 113–132, London, England, Sept 2004.

[6] Gábor Horváth, Miklós Telek, and Csaba Vulkán. AAL2 multiplexing delay calculations in UTRAN. In *Proceedings 11th Microcoll conference*, Budapest, 2003.

[7] Chunlei Liu, Sohail Munir, Raj Jain, and Sudhir Dixit. Packing density of voice trunking using AAL2. In *Proceedings IEEE Global Telecommunications Conference (GlobeCom99)*, volume 1(B), pages 611–615, Rio de Janeiro, Brazil, December 1999.

[8] R. Makké, S. Tohmé, J. Y. Cochennec, and S. Pautonnier. Performance of the AAL2 protocol within the UTRAN. *Annals of Telecommunications Journal*, 58/7-8, July-August 2003.

[9] Soracha Nananukul, Yile Guo, Maunu Holma, and Sami Kekki. Some issues in performance and design of the ATM/AAL2 transport in the UTRAN. In *IEEE Wireless Communications and Networking Conference*, Sept. 2000.

[10] C. G. Park, D. H. Han, and J. I. Jung. Interdeparture time analysis of CBR traffic in AAL multiplexer with bursty background traffic. *IEE Proceedings on Communications*, 148:310–315, October 2001.

[11] W. Petr, R. R. Vatte, P. Sampath, and Y. Lu. Efficiency of AAL2 for voice transport: Simulation comparison with AAL1 and AAL5. In *Proc. IEEE ICC.*, volume 2, pages 896–901, June 1999.

[12] H. Saito. Performance evaluation and dimensioning for AAL2 CLAD. In *Proc. IEEE lnfocom*, volume 1, pages 153–160, Mar. 1999.

[13] K. Sriram and Wang Yung-Terng. Voice over ATM using AAL2 and bit dropping: performance and call admission control. *IEEE Journal on Selected Areas in Communications*, 17:18 – 28, January 1999.

[14] Kotikalapudi Sriram and Ward Whitt. Characterizing superposition arrival processes in packet multiplexers for voice and data. *IEEE Journal on Selected Areas in Communications*, 4:833 – 846, September 1986.

[15] B. Subbiah and S. Dixit. ATM adaptation layer 2 (AAL2) for low bit rate speech and data: Issues and challenges. In *Proc. IEEE ATM Workshop*, pages 225–233, May 1998.