Erlang Analysis of Cellular Networks using Stochastic Petri Nets and User-in-the-Loop Extension for Demand Control

Rainer Schoenen, Halim Yanikomeroglu
Department of Systems and Computer Engineering, Carleton University, Canada

Abstract—Cellular networks face severe challenges due to the expected growth of application data rate demand with an increase rate of 100% per year. Over-provisioning capacity has been the standard approach to reduce the risk of overload situations. Traditionally in telephony networks, call blocking and overload probability have been analyzed using the Erlang-B and Erlang-C formulas, which model limited capacity communication systems without or with session request buffers, respectively. While a closed-form expression exists for the blocking probability for constant load and service, a steady-state Markov chain (MC) analysis can always provide more detailed data, as long as the Markov property of the arrival and service processes hold. However, there is a significant modeling advantage by using the stochastic Petri net (SPN) paradigm to model the details of such a system. In addition, software tool support allows getting numeric analysis results quickly by solving the state probabilities in the background and without the need to run any simulation. Because of this efficiency, the equivalent SPN model of the Engset, Erlang-B and Erlang-C situation is introduced as novelty in this paper. Going beyond the original Erlang scenario, the user-in-the-loop (UIL) approach of demand shaping by closed-loop control is studied as an extension. In UIL, demand control is implemented by a dynamic usage-based tariff which motivates users to reduce or postpone the use of applications on their smartphone in times of light to severe congestion. In this paper, the effect of load on the price and demand reduction is modeled with an SPN based on the classical Erlang Markov chain structure. Numeric results are easily obtained and presented in this paper, including probability density functions (PDF) of the load situation, and a parameter analysis showing the effectiveness of UIL to reduce the overload probability.

Keywords—User-in-the-loop (UIL); demand shaping; demand control; congestion; Erlang; stochastic Petri-net (SPN).

I. INTRODUCTION

In cellular networks the trend towards increasing data rates continues, with predictions of up to 100% increase per year. Figure 1 shows what this would mean for a system where the capacity cannot be raised by the same factor. At some point in time the capacity is exceeded by the demand (not the traffic, which will be choked by packet losses). This will happen at a different times for different cellular locations and is subject to daily fluctuations as well. As we are particularly concerned about the busy hours, i.e., the times where congestion is likely to happen, every broadband wireless access point will face this problem at some time. There is a well-known theory for the blocking probability of such scenarios [1]. The novel UIL paradigm goes beyond that and allows soft-CAC compared to hard-CAC (CAC is call admission control).

Recently, the suitability of the Erlang approach for Internet traffic has been validated [2] and there are still active publications in this area [3]. In the wireless context the situation is similar, given the limited capacity, which would only allow a few simultaneous high-definition video transmissions at a time in the same cell. The scenario assumes stationary users and quasi stationary capacity. The classical Erlang-B and C formulas provide a closed-form result for the blocking probability $P_b$ and the waiting probability $P_w$, respectively, with numeric complexity $O(C)$, but it does not provide advanced statistics such as probability mass functions (PMF) of the channel usage and does not allow any modification in the Markov chain.

Stochastic Petri nets (SPN) are known to generate Markov chains (MC) [4]. They have rarely been applied to communications problems yet, but few examples include communication networks [5], protocols [6], WiMAX [7], wireless scheduling [8], ad-hoc networks [9], radio channels [10] and flow control [11]. For a quick introduction on SPN refer to [12].

In this paper we present SPN models which are able to reproduce the Erlang results precisely, because they correspond to the same Markov chain. In addition, the SPN allows calculation of PMF, so that we are able further analyze the channel utilization and waiting statistics by specifying reward measures of interest. Tool-support is available [13] and makes the generation of result graphs a job of 10 seconds. Having the SPN models is a significant achievement, because it allows SPN methodology to be used to extend the theory beyond the classical Erlang results. In this paper the SPN is extended to include the user-in-the-loop (UIL) [14], [15] paradigm, where demand shaping changes the arrival rate of new sessions depending on the congestion status, i.e., the number of already active sessions by an anticipated dynamic price increase and demand decrease. Analysis results in this paper are obtained by direct numeric MC solution from the SPN, without any need for simulation. Results show how the blocking and waiting probability can be reduced significantly by the UIL method, by reducing excess traffic demand, thus limiting congestion in the network.

The paper organization is as follows. The classical Erlang reasoning and the SPN models are introduced in section II. Then the UIL concept and its SPN model are introduced. Next analysis results are provided before the paper ends with a conclusion.
II. ERLANG MODEL

This section provides SPN models which represent the known scenarios for Erlang-B, Erlang-C and Engset. As the reader will see, the SPN steady-state solution contains all the reward measures of interest, whereas the analytic solutions only provide mean values. Some require numeric iterations and summations, and are known as numerically instable.

The Erlang-B scenario models session arrivals as memoryless with exponential interarrival time $1/\lambda$, and a capacity of $C$ times the requirements per session ($C$ servers) [1]. Thus the $M/M/C$ queueing model and its Markov chain (MC) are applicable. Figure 2 shows the MC assuming $\Lambda_i = \lambda$ and no states beyond $C$. Equation 1 [1] provides the blocking probability $P_b$:

$$P_b = B(u, C) = \frac{\sum_{i=0}^{\min(C, u)} \frac{u^i}{i!}}{\sum_{i=0}^{C} \frac{u^i}{i!}}$$

$$u = \lambda \cdot h \quad \text{(in Erlangs)}$$

The SPN model of the Erlang-B scenario is shown in Figure 3. $T1$ and $T0$ are the generating and serving transitions. $P0$ is the supply ($n = S$ can generally be assumed very large compared to all other numbers, $S \gg C$). It is not relevant for the queueing model, but only for bounding the MC state space of the SPN. $P1$ represents the state of active sessions (e.g., currently carried video streams). It is limited by capacity $m = C$, thus the disabling arc to $T2$, which is an immediate transition without timing. If $T2$ cannot fire due to full capacity, $T3$ takes over, which has lower priority than $T2$. Thus for Erlang-B, $P2$ would never contain tokens in a tangible state (e.g., does not constitute a state of the MC). The state index $i$ is determined by $i = \#P1 + \#P0$, where $\#Px$ denotes the number of tokens in a place. The timing is defined as follows: $\lambda(T1) = u/h$ or $\tau(T1) = h/u$ specify the aggregate generation rate of sessions. The session have an average duration of $h$ seconds, thus each active session is served by $T0$ with $\tau = h$, therefore $\tau(T0) = h/\#P1 = h/i$ or $r(T0) = i \cdot h$ is the state-dependent server timing. In Figure 2, $\mu$ equals $h$.

The Engset scenario is like the Erlang-B scenario, but with limited customers ($n = S$). Its MC is Figure 2 assuming $\Lambda_i = (S-i) \cdot \lambda$ and no states beyond $C$. Its blocking probability exists in closed form, but requires recursion and many iterations to converge [1]:

$$P_b = \frac{\sum_{x=1}^{C} \left(\frac{(S-1)!}{x! \cdot (S-x-1)!}\right) \cdot M^x}{\sum_{x=1}^{C} \left(\frac{(S-1)!}{x! \cdot (S-x-1)!}\right) \cdot M^x}$$

$$M = \frac{\mu}{S - u \cdot (1 - P_b)}$$

The Engset case is included in the SPN of Figure 3 by reducing the initial tokens in $\#P0(t = 0)$ to $n = S$ (supply). In all other cases the supply is chosen well beyond the numbers of relevance.

The Erlang-C scenario differs from Erlang-B by the existence of a waiting buffer for sessions which cannot be carried at the moment, but are taken into account as soon as capacity becomes available again. This is typical for an application scenario where the user clicks on a video link, and it takes a few up to many seconds until the video really starts. Its MC is Figure 2 assuming constant $\Lambda_i = \lambda$ and infinitely many states beyond $C$. The service departure beyond state $C$ is constant $C \cdot \mu$, as this is the maximum capacity to serve the active sessions. The Erlang-C waiting probability is known as...
in Equation 5:

\[
P_w = \frac{u^C}{C!} \sum_{i=0}^{C-u} \frac{1}{i!} \frac{\lambda^i}{C!} (6)
\]

Figure 4 shows the SPN for the Erlang-C scenario. It is similar to Erlang-B in Fig. 3, but this contains a buffer \( P_2 \) which is not flushed at overload. Instead, if the capacity is exceeded, session requests wait in \( P_2 \) to be served soon. This makes up for a bit more load than Erlang-B, thus \( P_w \) is generally higher than \( P_b \).

Using SPN tools [13], the MC could be generated and solved very quickly (less than 10s). Figures 5, 6, 7 show numeric results of the MC analysis of the Erlang-C scenario. Results for Erlang-B are omitted due to space limitations. As can be seen, the PMF of tokens in place \( P_1 \) reveals the stochastic load distribution around the average load of \( u \) Erlangs, which has been studied for \( u \in [50...100] \). For a load of \( u = 75 \), a significant probability of overload (congestion) is visible, just below 1%. Once we are in congestion, unserved sessions wait in \( P_2 \), with a probability of exceeding any \( x \) given in Fig. 7. For \( u = 100 \) and beyond the system is not stable. In the next section a solution for levitating this congestion is introduced.

III. USER IN THE LOOP

The UIL paradigm is a shift from assuming user traffic as constant, given from outside of the system, towards assuming now this traffic (more precisely, the demand) can be influenced or shaped by the system itself. For wireless cellular communications there are two flavors of UIL, the spatial [17] by suggesting relocation to a point of better spectral efficiency and the temporal [14] by convincing users to

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig4.png}
\caption{SPN model for Erlang-C scenario. Compared to Fig. 3, there is no \( T_3 \) and thus \( P_2 \) serves as the waiting buffer for sessions.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig5.png}
\caption{Probability mass function (PMF, linear scale) of tokens in place \( P_1 \) which represents the number of active sessions. The same PMF is shown in Fig. 6 in logarithmic scale.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig6.png}
\caption{Probability mass function of tokens in place \( P_1 \) in logarithmic scale. \( P_1 \) models the number of sessions currently active and carried by the system. The accuracy \( 10^{-9} \) cannot be achieved so clearly by means of simulation. Artefacts at the rightmost position (\#P1 = 100) are correct and contain the sum of all probabilities which would lead to traffic beyond 100% and are waiting in \( P_2 \) instead.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig7.png}
\caption{Probability mass function of tokens in place \( P_2 \) which represent the number of sessions waiting for available capacity. Obviously \( u = 100\% \) is absolute overload.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig8.png}
\caption{User-in-the-loop (UIL): control of user and system [17]. Quantified User Information (QUI) in this paper is the indication of a dynamic price.}
\end{figure}
Most elements are equal to Figure 4, basically the loop corporate the UIL control aspect, as can be seen in Figure 11. The controller has to determine a price increase $\chi$ as

$$\pi = \pi^{(N)} \cdot (1 + \chi).$$

For the purpose of this paper, only one QoS class is assumed, which would be video because of its presumed dominance in the future. Figure 9 shows the traffic of an example week and the outcome of the UIL control.

The controller [16] has to determine a price increase $\chi$ as the incentive to the user, but internally installs a control ratio $p$.

Now the SPN model of section II is extended to incorporate the UIL control aspect, as can be seen in Figure 11. The main modification is $P3$ with $T3$, which has a (hidden) enabling function of $#P1 \leq u_T$, which flushes all tokens out of $P3$ as long as we are below the target threshold $u_T$. Above $u_T$, $P3$ holds tokens proportional to the severity of excess, $\#P3 = \Delta u = (i - u_T)$. The UIL controller calculates the dynamic price as $\chi = cp \cdot \Delta u$, using the proportionality ($P$) factor $c_p$. There is no integral (I) or differential (D) component here. With this $\chi$ we know the user reaction according to Fig. 10. Therefore, the demand would reduce from $u$ to

$$u \cdot e^{-p \cdot c_p \cdot \Delta u}$$

and this is installed by adjusting the generator rate to be

$$\tau(T1) = h/u \cdot e^{pc_p \cdot \Delta u}$$

The additional transitions $T4$ and $T5$ are there in order to empty place $P3$ in sync with $P1$, by setting priority levels $p(T3) > p(T2) > p(T5)$ so that

$$\#P3 = \max(0; #P1 - u_T)$$

Results for the UIL scenario are shown in Figure 12 and following. As written in Table II, the average demand load was set to $u = 90$ Erlangs. The parameter $c_f = c_p$ is the proportional control factor. For $c_p = 0$ there is no UIL control and results reflect the Erlang-C results. Thus Figure 12 is the CCDF of Figure 5 for $c_p = 0$ and $u = 90$. With stronger
TABLE II. UIL assumptions for the user and controller box

<table>
<thead>
<tr>
<th>Property</th>
<th>var</th>
<th>setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load threshold value (in SPN)</td>
<td>( p^w_t )</td>
<td>90%</td>
</tr>
<tr>
<td>Price increase</td>
<td>( u_T )</td>
<td>([0...2])</td>
</tr>
<tr>
<td>Exponential fit for user reaction model</td>
<td>( p^{exp}(\chi) )</td>
<td>( e^{-\eta \cdot \chi} )</td>
</tr>
<tr>
<td>Elasticity (log) [19]</td>
<td>( \eta )</td>
<td>1.3</td>
</tr>
<tr>
<td>Control factor (P as of PID)</td>
<td>( c_p )</td>
<td>([0...0.1])</td>
</tr>
<tr>
<td>Average load for Fig. 12 and following</td>
<td>( u )</td>
<td>90</td>
</tr>
</tbody>
</table>

Fig. 12. Effect of UIL control on the number of sessions active, which is depicted by the CCDF \( #P_1 \). We observe that only sessions beyond the threshold of 90% are controlled down.

Control factors up to \( c_p = 0.1 \), the probability of blocking or waiting is reduced from 0.2 to \( 10^{-4} \). Figure 13 shows what happens in overload situations, as tokens in \( P_2 \) represent (video) sessions waiting for capacity. Without UIL control, there is a significant number of sessions unserved. Even 100 unserved sessions (while 100 are served) are possible with probability in the order of \( 10^{-5} \). The graph drops at \( #P_2 = 100 \) only because of the limited supply of 200 sessions, but a logarithmic extrapolation is possible. With UIL control, \( Pr(#P_2 > x) \) drops to very low numbers, as expected. Figure 14 is basically a zoom into Fig. 12 due to Eq. 10, but it can be observed how precisely the UIL control “bends” the demand above the threshold. The following figures show scalar results by varying the control factor. In Figure 15 the probability of exceeding the target threshold is studied. Naturally, as \( u = 90 \) and (independently) \( u_T = 90 \) was chosen, this probability if 50% without UIL control. Using Little’s formula, the average waiting time was determined and shown in Figure 16. There is basically no waiting for \( c_p = 0.1 \), as instead some users decided not to watch the video in the current overload situation. Figure 17 displays how likely the capacity is exceeded in the given scenario of average load 90%. 20% is a relatively high number of users who would be frustrated not being able to use the service. Instead, with UIL, a comparable number of users would not use the service, but for a different reason: Well informed that this is a congestion situation, and sorted by willingness to pay more, i.e., the more urgent use case is preferred compared to the less important application.

IV. Conclusion

In this paper stochastic Petri net models for the Erlang-B and Erlang-C traffic scenarios are presented, as well as an extension to incorporate UIL demand shaping. As can be observed, the modeling efficiency of SPN allows modifying the underlying Markov chain by simple means of (functional) parameters of the SPN model. In addition, tool-supported Markov chain analysis does not require simulations and naturally delivers accuracies in the order of \( 10^{-9} \) or better within a few seconds of run time. It is also easy to obtain higher order statistics, e.g., PMF, CDF and CCDF graphs without any extra effort, because the steady-state Markov chain contains all the information already. The case of UIL analyzed and discussed in this paper shows how demand control can be incorporated into networks and reduces the overload probabilities significantly, compared to the Erlang-C scenario. Especially in wireless networks the capacity is assumed to be in congestion more and more often in the future. As an outlook, fading channel...
Fig. 15. The probability of traffic exceeding the threshold of 90% equals the reward measure $P_r(\#P_1 > 90)$.

Fig. 16. Average waiting time (in seconds) for free capacity, depending on the UIL control factor ($0 = \text{no UIL control}$).

models and capacity fluctuations due to user mobility can be incorporated into transition $T_0$. The UIL principle remains functional in this case.

REFERENCES