December 2006

Proposed Techniques to Dimension a Push-To-Talk over Cellular Server

Muhammad T. Alam
Bond University, Muhammad_Alam@bond.edu.au

Zheng da Wu
Bond University, Zheng_Da_Wu@bond.edu.au

Proposed Techniques to Dimension a Push-To-Talk over Cellular Server

M. T. Alam, *student member, IEEE* and Z. D. Wu, *member, IEEE*
School of IT, Bond University
Gold Coast, Australia
Fax: +61 7 55953320
malam@bond.edu.au

Introduction

Push-To-Talk can be viewed as an Instant Messaging service, enhanced with voice functionality. Most of the related work available today focuses on the performance analysis over PoC. However, they seemed to be completely ignorant about dimensioning PoC controller to optimize revenue for service providers. The contribution of this proposed work is as follows:

1. Provide access priority to special sessions based on available TRUs (Transmit/Receive Unit);
2. Optimize the session timer for a PoC controller;
3. Optimize number of session initiation for a PoC client during busy hour.

We propose to dimension the PoC controller based on the assumption that the network grade of service is provided. This way a PoC server is able to control PoC functionalities to the optimal level.

Proposed Access Priority Model:

PoC usage in GPRS has two main scenarios: 1. Short interactive sessions (type 1) and 2. Long session with sporadic, interactive talk periods (type 2). The distinction between the two talk is that one contains chat sessions after long intervals within a single session where as the other refers to the separate sessions for each talk. The key challenge is to reduce the session set up time. Because, in the session set up the steps to be performed are a) Paging with which the PoC server defines the location of the PoC terminal on cell level, b) Cell update with which the terminal tells the PoC server in which cell it is located c) Radio resource assignment procedures which are the part of session set up procedure and finally d) PoC signalling. Obviously the long sessions will prefer a pre-established session than on demand session set up. We define the access priority of these two kinds of session set up. Priority is provided to on demand session set up based on number of available TRUs.

Type 2 (pre established) sessions should not be allowed during the busy hour where as type 1 (on demand) sessions should be able to use any free TRU. Let, a Type 2 session can use a time slot only when the total number of busy TRU is less than some protection level of number *b*.

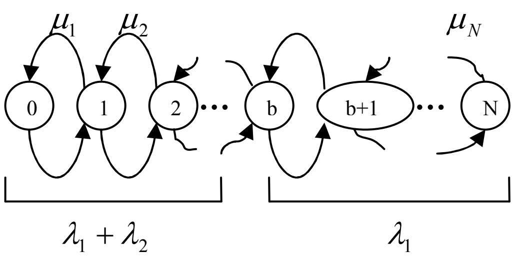

Figure 1-Markov model for accessing session

The corresponding Markov model state change with probabilities is presented in figure 1. A state *k* represents the number of sessions present in the PoC server. $\lambda_1, \lambda_2,\ and\ \mu_k$ are arrival and service rates of type 1 and type 2 sessions at state *k* respectively. Based on the above Markov model it is possible to compute the blocking probability of sessions and the desired threshold level.

Proposed Timer

The literature review suggests that a typical PoC session should not exceed 40 sec in rush hour. Our objective in this section is to control lifetime of the PoC sessions for a PoC controller. Define,

q(x) = The probability that *x* number of times a PoC session goes through a time slot of a TRU during time interval *T*

p = The probability all time slots are occupied during the interval *T*

t = Duration of a time slot (we use 20 ms which is practical in GPRS)

N = Total number of TRUs

We find,

$$p \approx \sum_{x=2}^{\infty} q(x) \qquad (1)$$

Given that a session is active during the whole interval *T*, the Poisson distribution *q(x)* is:

$$q(x) = \frac{(tT)^x}{x!} e^{-\left(t+\frac{1}{\mu}\right)T} \qquad (2)$$

A session may go through any of the *N* TRUs in a PoC server. Therefore,

$$q(x) \le \frac{(tT)^x}{x!} e^{-\left(t+\frac{1}{\mu}\right)T} \left(\frac{1}{N}\right) \qquad (3)$$

From Eq. (3) and (1) we get,

$$p \le \sum_{x=2}^{\infty} \frac{(tT)^x}{x!} e^{-\left(t+\frac{1}{\mu}\right)T} \left(\frac{1}{N}\right) \qquad (4)$$

Using the Taylor series

$$e^{(tT)} = 1 + tT + \frac{(tT)^2}{2!} + \ldots.. \qquad (5)$$

We find,

$$p \le \frac{e^{tT} - tT - 1}{Ne^{\left(t+\frac{1}{\mu}\right)T}} \qquad (6)$$

The above expression provides a relation between blocking probability and the session timer. We will further simplify and test the expression for a PoC server to provide to each PoC terminal during busy hour.

Proposed model to optimize simultaneous sessions

Our objective in this section is to control the number of simultaneous sessions for a PoC client during busy hour. Since, the Northstream report suggests that cost analysis based on time slots of PoC servers produce equal outcomes as that of TRUs, we consider our next analysis based on number of time slots. Gilbert's model (1960) have shown that a simple two-state Markov chain can measure packet loss over the internet efficiently. We use similar approach to compute the number the optimal sessions for a PoC client.

The two state natures of figure 2 and figure 3 can capture the bursty nature of the number of simultaneous sessions in busy hour. The model in figure 2 has two states: Blocking or busy and Not busy. H_1 and H_2 are the state transition probabilities.

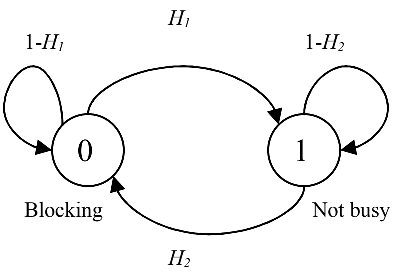

Figure 2-Markov model for PoC server states

The PoC server goes to Blocking state ***0,*** when all channels/time slots are busy at a random point of time that can be computed from Erlang's loss formula. In this state, number of session arrival in the server is greater than $5N_T$ (where N_T is the total number of time slots), assuming that a time slot serves 5 PoC sessions at the same time on the average.

$$H_2 = \frac{\frac{a^{N_T}}{N_T!}}{\sum_{d=0}^{N_T} \frac{a^d}{d!}} \qquad (7)$$

Where, a = total traffic offered

Here, we assume the grade of service, H_2 is provided. It goes to Not busy state ***1,*** when there is at least one time slot available that can be computed from the Binomial distribution. Any new session will be blocked when the server is in state ***0.*** A successful session set up only depends on the current state. Because of the throttled nature of the PoC sessions, a session changes between idle (inactive) and busy (active), the offered traffic per session is (let a, λ, μ are total traffic offered, total arrival rate and total service rate respectively)

$$\alpha = \frac{T_{busy}}{T_{idle} + T_{busy}} = \frac{1/\mu}{1/\lambda + 1/\mu} = \frac{a}{1+a} \qquad (8)$$

In fact, the statistical analysis shows that the voice activity factor has found to be 67%. That means that 33% of a conversation is actually pauses and silence.

Then, for non busy state,

$$H_1 = \sum_{d=0}^{N_T - 1} \binom{N_T}{d} \alpha^d (1-\alpha)^{N_T - d} \qquad (9)$$

The transition between two states occurs at each session set up/termination. Thus in steady state:

$$P(0) + P(1) = 1 \qquad (10)$$

The state transition matrix is given by

$$P_H = \begin{bmatrix} 1-H_1 & H_1 \\ H_2 & 1-H_2 \end{bmatrix} \qquad (11)$$

The session Blocking is equal to the state probability $P(0)$. Similarly the probability of successful session set up is equal to the state probability $P(1)$. Figure 3 represents the nature of a session initiation situation of a PoC client. State A represents a client initiating one session and state B represents multiple session initiation. I_1 and I_2 are the transition probabilities. The probability that a PoC client initiates a session is the mean arrival rate of all PoC clients i.e.,

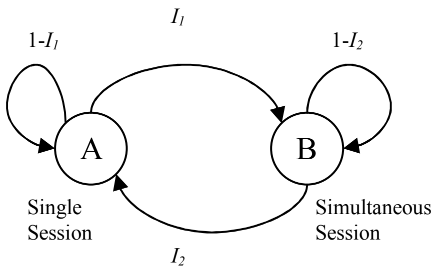

Figure 3-Session states of a PoC Client

$$I_2 = \frac{\lambda}{N_c} = \frac{\sum_{i=1}^{N_c} \lambda_i}{N_c} \qquad (12)$$

Where, N_c = The number of PoC clients being served by a PoC server.

The probability of simultaneous session initiation of a PoC client during a known period T can be determined by:

$$I_1 = 1 - \Pr[one\, session \mid T = t_s] = 1 - \frac{\lambda}{N_c} t_s e^{-\left(\frac{\lambda}{N_c} t_s\right)} \qquad (13)$$

Where, t_s is the session lifetime of a PoC client.

$$P(A) + P(B) = 1 \qquad (14)$$

The state transition matrix is:

$$P_I = \begin{bmatrix} 1-I_1 & I_1 \\ I_2 & 1-I_2 \end{bmatrix} \qquad (15)$$

Since the PoC server going to busy state depends on total number of sessions, we propose to concatenate two models to compute the simultaneous sessions which will lead the PoC server to the optimal value.

Conclusion

In this paper, we proposed several optimal characteristics to dimension a PoC server. Deployment requires optimized expertise in the entire service delivery chain from a cost point of view. A service provider can benefit from the models proposed in this paper. The models will be fine-tuned and a test-bed will be implemented in IP Multimedia Subsystem (IMS) environment to test the models in order to provide efficient PoC services to the IMS terminals. Also, we are currently investigating the case of prioritizing and classifying PoC traffic in terms of session dropping probabilities.

References

[1] Northstream AB (2004). "Overview and comparison of Push-to-talk" www.northstream.se
[2] Kim P., Balazs A., Broek E., Kieselmann G., Bohm W. (2005). "IMS-based Push-to-Talk over GPRS/UMTS" IEEE Wireless Communications and Networking Conference, Vol 4, pp: 2472-2477.
[3] Gilbert, E. N. (1960). "Capacity of a burst-noise channel," Bell Syst. Tech. J., vol. 39, pp: 1253-1265.
[4] DaSilva L. A., Morgan G. E., Bostian C. W., Sweeney D. G., Midkiff S. F., Reed J. H. and Thompson C. (2006). "The Resurgence of Push-to-Talk Technologies" IEEE Communication Magazine, pp:48-55.