Опис рачуна Ерлангове Б и Енгсетове формуле

Вељковић Срђан

Новембар 2010.

1 Увод

Ерлангова Б формула одређује губитке за неки понуђени саобраћај на одређеном броју "линија" (односно "носилаца саобраћаја").

Практичније, Ерлангова Б формула за одређен број (телефонских) позива, који треба да прођу преко одређеног броја говорних путева (канала), даје вероватноћу да позив неће моћи да прође (него ће да "падне").

Ерлангова Б формула претпоставља бесконачан број извора саобраћаја, као и да позив који је пао не утиче на даљи ток. Обе претпоставке су грубе.

Прва претпоставка је прикладна када је број извора значајно већи од броја канала (помињу се бројеви "преко 8 пута већа", мада нама није познато да је то математички изведена цифра). Ако то није случај, примењује се Енгсетова формула, коју обрађујем у одељку 3.

Друга претпоставка је прикладна ако су губици мали. Ако губици нису мали, онда се примењује "проширена Ерлангова Б формула", односно алгоритам. Приказ овог алгоритма дајемо у одељку 2.5.

За детаљнији опис може да се прочита неки напис о теорији саобраћаја.

1.1 Где је Б ту је и Ц

Постоји и Ерлангова Ц формула, која се односи на случај када позив, у случају да не успе, "стоји на чекању", док се не ослободи неки канал. Груба претпоставка је да стоји "до у бесконачност", ако то није испуњено, примењују се друге формуле које узимају у обзир "колико времена позив највише чека" или "колико позива може да чека истовремено", или оба.

Овде се даље не бавимо Ерланг Ц формулом и њеним сродницима.

2 Ергланг Б формула

Нажалост, не постоје неке прихваћене ознаке појмова у теорији саобраћаја, па ћемо користити своје. Дакле, Ерлангова Б формула је:

$$p_g = \frac{\frac{s^k}{k!}}{\sum_{i=0}^{k} \frac{s^i}{i!}} \tag{1}$$

где је:

p_g вероватноћа губитка позива

s понуђени саобраћај у Ерланзима (број позива у једном тренутку)

k број канала за пренос саобраћаја (позива)

2.1 Рачун

Пошто Ерланг Б формула користи факторијел, јасно је да је примена формуле "тако како стоји" неизводљива у данашњим рачунарима за иоле веће вредности k, јер ће вредност факторијела или изаћи из опсега, или ће успорити обраду до бесмисла.

Сем тога, Ерлангова Б формула практично даје зависност три параметра. Често је занимљиво одредити неки од два улазна параметра Ерлангове Б формуле, на основу оног другог улазног и резултата. Дакле, ако знамо понуђени саобраћај, можемо да одредимо број потребних канала за одређену вероватноћу губитака. Такође, ако нам је наметнут број канала, можемо за одређену вероватноћу губитака да одредимо колики понуђени саобраћај можемо да прихватимо.

2.2 Рекурзивни рачун Ерланг Б формуле

Овај рачун је прилично познат. Идеја је да, за исти саобраћај, можемо вредност за дати број канала да одредимо на основу вредности за претходни број канала (за један мањи). Ради прегледности ћемо рачунати инверзну вредност.

$$\frac{1}{p_g} = \frac{\sum_{i=0}^{k} \frac{s^i}{i!}}{\frac{s^k}{k!}} = \frac{k!}{s^k} \sum_{i=0}^{k} \frac{s^i}{i!} = \sum_{i=0}^{k} \frac{k! s^k}{s^i i!}$$

$$\frac{1}{p_g} = \sum_{i=0}^{k} \frac{k!}{s^{k-i} i!} \tag{2}$$

$$\frac{1}{p_g(k+1)} = \sum_{i=0}^{k+1} \frac{(k+1)!}{s^{k+1-i} i!} = \frac{(k+1)!}{(k+1)!} + \sum_{i=0}^{k} \frac{(k+1)!}{s^{k+1-i} i!} = 1 + \frac{k+1}{s} \sum_{i=0}^{k} \frac{k!}{s^{k-i} i!}$$

$$\frac{1}{p_g(k+1)} = 1 + \frac{k+1}{s} \frac{1}{p_g(k)} \tag{3}$$

Почетна вредност - за $k = 0$, се добија простим рачунањем формуле 2:

$$\frac{1}{p_g(0)} = \frac{0!}{s^0 0!} = 1$$

Изражено алгоритамски, у програмском језику Руби:

```
def erlangb(s, k)
    r = 1.0
    1.upto(k) { |j| r = 1 + r*j/s }
    return 1.0 / r
end
```

Наравно, изворни код за "индустријску примену" би требало да садржи и провере опсега и сл.

2.3 Одређивање броја канала методом половљења

За одређивање броја канала, приметимо да је, за непроменљив саобраћај, p_g равномерно опадајућа функција (како k расте, тако p_g опада). Ово је интуитивно јасно, што је више канала, то ће губици за исти саобраћај бити мањи. Ипак, ово може и математички да се докаже.

Овде дајемо како су то извели Кјао и Кјао, у раду који је (док ово пишемо) јавно доступан на Интернету на адреси:

`http://www.cas.mcmaster.ca/~qiao/publications/erlang/newerlang.html`

Занимљиво је приметити да Кјао и Кјао дају другачије одређивање рекурзивне Ерланг Б формуле, које даје другачији резултат од "нашег". Како нису дали детаљан поступак, није јасно да ли су погрешили у поступку, или су разлике занемарљиве у опсегу који је занимљив у примени.

Ипак, њихов доказ да p_g равномерно опада за непроменљив саобраћај је ваљан. Овде га "препричавамо":

$$p_g(k+1) = \frac{\dfrac{s^{k+1}}{(k+1)!}}{\displaystyle\sum_{i=0}^{k+1} \frac{s^i}{i!}} = \frac{s}{k+1} \frac{\dfrac{s^k}{k!}}{1 + \displaystyle\sum_{i=1}^{k+1} \frac{s^i}{i!}} < \frac{\dfrac{s^k}{k!}}{\dfrac{k+1}{s} \displaystyle\sum_{i=1}^{k+1} \frac{s^i}{i!}} = \frac{\dfrac{s^k}{k!}}{\dfrac{k+1}{s} \displaystyle\sum_{i=0}^{k} \frac{s^{i+1}}{(i+1)!}} =$$

$$= \frac{\dfrac{s^k}{k!}}{\displaystyle\sum_{i=0}^{k} (\frac{k+1}{i+1} \frac{s^i}{i!})} < \frac{\dfrac{s^k}{k!}}{\displaystyle\sum_{i=0}^{k} \frac{s^i}{i!}} = p_g(k)$$

Укратко :

$$p_g(k+1) < p_g(k)$$

Ако знамо да је функција монотоно опадајућа, онда можемо да применимо метод половљења. Требају нам само две вредност $p_g(k)$, једна мања и једна већа од улазног P_g - односно две вредности k за које то важи.

Логично је да почнемо од 0 и понуђеног саобраћаја као почетних вредности. Али, ако је $p_g(s) < P_g$, онда треба да нађемо такав $k > s$ за који је $p_g(k) > P_g$.

Кјао и Кјао предлажу, без образложења (сем “закључило смо да је тако најбоље”) да се овде понуђени саобраћај повећава са 32, док се не нађе то што тражимо. Пошто није јасно како ће “један те исти” број да буде добар за различите вредности понуђеног саобраћаја, то 32 у овом случају чини магијским бројем. Како смо одлучно против магијских бројева, не свиђа нам се такво решење.

Сматрамо да је смисленије применити неки фактор, уместо броја, јер ће то потрагу учинити зависном од саобраћаја. Сматрамо да је фактор 1,5 прикладан. Наиме, губици морају да буду врло мали да буде потребно 50% више канала него што је понуђени саобраћај, што се у пракси ретко среће. Ако се ипак и деси, после неколико пролаза би требало да завршимо нашу претрагу.

Сада можемо да прикажемо и алгоритам, у Рубију:

```
def kanala_erlangb(p_g, s)
    l, r = 0, s.ceil
    while erlangb(s, r) > p_g
        l = r
        r = 1 + (r * 3) / 2
    end
    mid = 0
    while (r-l) > 1
        mid = (l+r+1)/2
        if erlangb(s, mid) > p_g
            l = mid
        else
            r = mid
        end
    end
    return r
end
```

Опет да приметимо да код за “стварни свет” треба да садржи и провере опсега, а може да садржи и неке оптимизације (рецимо, ако је такве среће да “нађе” P_g, може одмах да прекине тражење).

2.4 Одређивање могућег саобраћаја методом половљења

За одређивање саобраћаја, приметимо да је, за непроменљив број канала, p_g равномерно растућа функција (како s расте, тако p_g расте). Ово је интуитивно јасно, што је већи саобраћај, то ће губици за исти број канала бити већи. Ипак, ово може и математички да се докаже.

Овде примењујемо класичан метод, употребом првог извода. Доказаћемо:

$$p_g'(s) > 0 \iff k = const \tag{4}$$

Применом познатог идентитета:

$$f'(\frac{u}{v}) = \frac{f'(u)f(v) - f(u)f'(v)}{v^2}$$

на формулу 1, под претпоставком да је k константно, добијамо:

$$p_g'(s) = \frac{\frac{ks^{k-1}}{k!}\sum_{i=0}^{k}\frac{s^i}{i!} - \frac{s^k}{k!}\sum_{i=0}^{k}\frac{is^{i-1}}{i!}}{(\sum_{i=0}^{k}\frac{s^i}{i!})^2}$$

За наше потребе је довољно да приметимо да је испод разломачке црте квадрат неког израза, што значи да је ≥ 0, а може да буде 0 само ако је $s = 0$ (нема саобраћаја), што није случај који нам је занимљив. Дакле, можемо да се ограничимо на анализу знака онога изнад разломачке црте.

$$\frac{ks^k}{k!s}\sum_{i=0}^{k}\frac{s^i}{i!} - \frac{s^k}{k!}\sum_{i=0}^{k}\frac{is^{i-1}}{i!} = \frac{s^k}{k!}(\sum_{i=0}^{k}\frac{ks^{i-1}}{i!} - \sum_{i=0}^{k}\frac{is^{i-1}}{i!})$$

Како је $\frac{s^k}{k!} > 0$, можемо да се ограничимо на знак онога што остане у загради:

$$\sum_{i=0}^{k}\frac{ks^{i-1} - is^{i-1}}{i!} = \sum_{i=0}^{k}\frac{s^{i-1}(k-i)}{i!}$$

Одавде је јасно да знак зависи само од знака израза:

$$\sum_{i=0}^{k}(k-i) = (k+1)k - \sum_{i=0}^{k} i = k(k+1) - \frac{k(k+1)}{2} > 0$$

Надаље је поступак сличан поступку за одређивање броја канала, па без даљих објашњења дајемо Руби код.

```
def saob_erlangb(p_g, k)
    l, r = 0, k.ceil
    while erlangb(s, r) < p_g
        l = r
        r = 1 + (r * 3) / 2
    end
    mid = 0
    while (r-l) > 1
        mid = (l+r+1)/2
        if erlangb(mid, k) < p_g
            l = mid
        else
            r = mid
        end
    end
    return r
end
```

2.5 Алгоритам проширене Ерлангове Б формуле

Ако са p_d означимо вероватноћу да позив који не успе буде (одмах) поновљен, алгоритам је:

1. Израчунај Ерланг Б формулу 1
2. Увећај понуђени саобраћај за: sp_dp_g
3. Израчунај Ерланг Б формулу са саобраћајем добијеним у претходном кораку
4. Понављај претходна два корака док p_g не постане стабилно.

Под “постане стабилно” се подразумева да промене резултата (p_g) услед примене овог алгоритма постану занемарљиве.

Изражено у Рубију:

```
def erlangb_prosxiren(s, k, p_d, epsilon)
    r, p_g = 0.0, erlangb(s, k)
    delta = epsilon
    while delta >= epsilon
        r = erlangb(s, k)
        delta = (r - p_g).abs
        p_g = r
        s += s * p_d * p_g
    end
    return p_g
end
```

У коду је `epsilon` граница разлике између два узастопна резултата Ерлангове Б формуле испод које се сматра да је алгоритам извршен.

Рачун броја канала или максималног саобраћаја по проширеној Ерланговој Б формули (алгоритму) је у основи исти као рачун по (основној) Ерланговој Б формули. Само је додатак што се при рачунањима Ерлангове Б формуле у току примене метода половљења користи и p_d. Толико је једноставно да га овде нећемо приказивати.

3 Енгсетова формула

У извесном смислу Енгсетова формула је општија Ерланг Б формула, која не занемарује број извора позива.

$$p_g = \frac{s^k \begin{pmatrix} v \\ k \end{pmatrix}}{\sum\limits_{i=0}^{k} s^i \begin{pmatrix} v \\ i \end{pmatrix}} \tag{5}$$

У односу на Ерлагову Б формулу (1), овде имамо нову величину v која представља број извора позива ($v \geq k$).

Еквивалентан запис ове формуле:

$$p_g = \frac{\dfrac{s^k v!}{(v-k)!k!}}{\displaystyle\sum_{i=0}^{k} \frac{s^i v!}{(v-i)!i!}} = \frac{\dfrac{s^k}{(v-k)!k!}}{\displaystyle\sum_{i=0}^{k} \frac{s^i}{(v-i)!i!}}$$

Сада је очигледно да су Енгсетова и Ерлангова Б формула врло сличне. Услед тога, на врло сличан начин долазимо до рекурзивне формуле за рачунање.

$$\frac{1}{p_g(k)} = \frac{(v-k)!k!}{s^k} \sum_{i=0}^{k} \frac{s^i}{(v-i)!i!}$$

$$\frac{1}{p_g(k+1)} = \frac{(v-k-1)!(k+1)!}{s^{k+1}} \sum_{i=0}^{k+1} \frac{s^i}{(v-i)!i!} =$$

$$= \frac{(v-k-1)!(k+1)!}{s^{k+1}} \left(\sum_{i=0}^{k} \frac{s^i}{(v-i)!i!} + \frac{s^{k+1}}{(v-k-1)!(k+1)!} \right)$$

$$= \frac{(k+1)}{s(v-k)} \frac{(v-k)!k!}{s^k} \sum_{i=0}^{k} \frac{s^i}{(v-i)!i!} + 1 = \frac{(k+1)}{s(v-k)} \frac{1}{p_g(k)} + 1$$

Очигледно смо и добили врло сличну формулу, али ћемо је мало променити да би била јаснија за кодирање:

$$\frac{1}{p_g(k+1)} = \frac{(k+1)}{s(v+1-(k+1))} \frac{1}{p_g(k)} + 1 \tag{6}$$

Као и за Ерланг Б формулу, и овде непосредном заменом добијамо:

$$p_g(0) = \frac{s^0 v!}{s^0 v!} = 1$$

Одатле се лако добија Руби код:

```
def engset(s, k, v)
    r, v_1 = 1.0, v+1
    1.upto(k) { |j| r = 1 + r*j/(s*v_1) }
    return 1.0 / r
end
```

Овде је занимљиво приметити да је за рачун Енгсетове формуле "популарнији" другачији метод, који чак не решава проблем великих факторијела. Могуће је да је то последица тога што се Енгесетова формула углавном примењује за мале бројеве, за које то и није проблем. Ипак, сматрамо да је овај рачун бољи јер омогућава примену Енгсетове формуле и случајевима не тако малих бројева, за случај када извори позива праве велики саобраћај. То је погодно за рачунања "најгорег случаја" при пројектовању система.

Рачун броја канала, извора и максималног саобраћаја када су познати остали параметри се ради слично као за Ерланг Б формулу. Разматрања на ту тему ће можда накнадно бити додата овде, ако се укаже потреба.