Lecture 20: Queueing Systems - Erlang Fixed Point Method

Concepts checklist

At the end of this lecture, you should be able to:

- *state* the Erlang fixed point method, and a numerical procedure to implement it.

Example 14 - Two link, slightly more complex routing, continued

Let us use an approximate technique by assuming now that

- link 1 is blocked with probability α_1, and
- link 2 is blocked with probability α_2.

Then, the demand made on each link using our previous reduced load approach is:

Route	Link 1	Link 2
AB	0.5	0
BC	0	0.6
AC	$0.3(1-\alpha_2)$	$0.3(1-\alpha_1)$
Total	$y_1 = 0.8 - 0.3\alpha_2$	$y_2 = 0.9 - 0.3\alpha_1$

$$\alpha_1 = B(1, y_1) = \frac{y_1}{1+y_1} = \frac{0.8-(0.3)\alpha_2}{1.8-(0.3)\alpha_2}$$

$$\text{and} \quad \alpha_2 = B(1, y_2) = \frac{y_2}{1+y_2} = \frac{0.9-(0.3)\alpha_1}{1.9-(0.3)\alpha_1}.$$

Solving these equations gives (the only sensible solution) as

$$\alpha_1 = 0.4007 \qquad \text{and} \qquad \alpha_2 = 0.4381.$$

Approximate blocking probabilities for the routes can now be calculated and compared with the exact solutions evaluated above.

Blocking on route AC is approximated by $1-(1-\alpha_1)(1-\alpha_2) \approx 0.6633$.

Route	AB	BC	AC
Exact	0.4074	0.4444	0.6296
Approximate	0.4007	0.4381	0.6633

The accuracy of the approximation is significantly better than that which we saw in the previous simpler example. Generally, the more complex the routing and the network, the more accurate the approximation becomes!

The idea behind the EFPM approximation is ...

- For each link j, let α_j be an approximation of the probability that an incoming request for a single circuit on link j is blocked.
- Calls for routes through link j are only accepted if **all** the links on the route are available.
- Consequently, not all requests for circuits on link j can be accepted, even if the link has sufficient circuits available. For example, with a route r through j, a request may be "rejected" because the circuit required on a different link i of r is not available. The probability of this is approximately α_i.

So...

- In order to compensate for the unavailability of the other circuits, the reduced load of requests for circuits on link j is used, instead of the actual load of requests for link j (which is simply $\sum_{\{r:j\in r\}} a_r$).
- If requests for circuits were to arrive as a Poisson process with rate y_j, then
  1. the link blocking probability would be given by $\alpha_j = B(c_j, y_j)$
  2. and the loss probability on route r would satisfy $B_r = 1 - \prod_{j\in\mathcal{J}}(1-\alpha_j)^{A_{j,r}}$ exactly, as a call is accepted if and only if it is accepted on all links.

We now give a mathematical description of the EFPM, which is most commonly used in practice in the telecommunications industry.

Definition 18 (Erlang Fixed Point Method)**.** *For a network comprised of a set of links $j \in \mathcal{J}$, let $\{\alpha_j, j \in \mathcal{J}\}$ be the unique solution to the equations*

$$\alpha_j = B(c_j, y_j), \text{ where } y_j = \sum_{\{r:j\in r\}} a_r \prod_{\{i\in r, i\neq j\}} (1-\alpha_i)$$

- *a_r is the offered load on route r and*
- *$B(\cdot,\cdot)$ is the Erlang Loss Formula.*

The vector $\boldsymbol{\alpha} = (\alpha_j, j \in \mathcal{J})$ is called the Erlang fixed point solution and an approximation for the loss probability on route r is given by,

$$B_r \approx 1 - \prod_{j\in\mathcal{J}}(1-\alpha_j)^{A_{j,r}} \ ,$$

where $A_{j,r}$ is the number of circuits that route r calls use on link j (here, either 0 or 1)

The Erlang fixed point approximation usually works well if the number of circuits and the associated demands are very high, or if the network is such that the routing is highly diverse. This means that on any link the circuits in use are likely to be from different routes and so likely to be connected via different links.

It is possible to improve the accuracy of the approximation, while still using reduced load techniques, by assuming that larger *modules* of the network, rather than individual links, operate "independently" of each other. Of course, this will mean that each module is larger and hence more difficult to analyse. However, careful choice of the modules may lead to significantly more accurate results with only a slight increase in complexity.

An example in which the approximation procedure should not be expected to perform well is where a number of small capacity links are arranged one after another in a line like our initial example. Here we would expect considerable dependence between the number of free circuits on adjacent links.

Also note that our model assumes that routes are fixed.

Example 11 (continuing from Lecture 18)

$$\begin{aligned}\Pr(\text{a call on route } \mathrm{O}_1 - \mathrm{D}_1 \text{ is accepted}) &= 1 - \Pr(\text{a call on route } \mathrm{O}_1 - \mathrm{D}_1 \text{ is blocked}) \\ &= \frac{G(\boldsymbol{c} - A\boldsymbol{e}_1, R)}{G(\boldsymbol{c}, R)} \\ &= \frac{G(\boldsymbol{c} - A\boldsymbol{e}_1, 4)}{G(\boldsymbol{c}, 4)}\end{aligned}$$

where

$$\begin{aligned}G(\boldsymbol{c}, 4) &= \sum_{\{\boldsymbol{n}:A\boldsymbol{n}\le\boldsymbol{c}\}} \prod_{r=1}^{4} \frac{\lambda_r^{n_r}}{n_r!} = \sum_{\{\boldsymbol{n}:A\boldsymbol{n}\le\boldsymbol{c}\}} \frac{60^{n_1}}{n_1!}\frac{70^{n_2}}{n_2!}\frac{50^{n_3}}{n_3!}\frac{80^{n_4}}{n_4!}, \\ G(\boldsymbol{c} - A\boldsymbol{e}_1, 4) &= \sum_{\{\boldsymbol{n}:A\boldsymbol{n}\le\boldsymbol{c}-A\boldsymbol{e}_1\}} \prod_{r=1}^{4} \frac{\lambda_r^{n_r}}{n_r!} = \sum_{\{\boldsymbol{n}:A\boldsymbol{n}\le\boldsymbol{c}-A\boldsymbol{e}_1\}} \frac{60^{n_1}}{n_1!}\frac{70^{n_2}}{n_2!}\frac{50^{n_3}}{n_3!}\frac{80^{n_4}}{n_4!}.\end{aligned}$$

The general formula for the Erlang fixed point equations is:

$$\alpha_j = B(c_j, y_j)$$

where

$$y_j = \sum_{r:j\in r} a_r \prod_{i\in r, i\ne j} (1 - \alpha_i),$$

a_r is the offered load for route r, and

$$B(c_j, y_j) = \frac{y_j B(c_j - 1, y_j)}{c_j + y_j B(c_j - 1, y_j)}.$$

Note that $B(c_j, y_j)$ is the Erlang-B formula, i.e. the probability that a call is lost on a link with c_j circuits and offered load y_j.

So, the eight equations for α_j are

$$\begin{aligned}&\alpha_1 = B(200, y_1), \quad \alpha_2 = B(100, y_2), \quad \alpha_3 = B(100, y_3), \quad \alpha_4 = B(100, y_4), \\ &\alpha_5 = B(100, y_5), \quad \alpha_6 = B(100, y_6), \quad \alpha_7 = B(100, y_7), \quad \alpha_8 = B(200, y_8),\end{aligned}$$

where

$$
\begin{aligned}
y_1 &= 60(1-\alpha_4)(1-\alpha_7) + 70(1-\alpha_5)(1-\alpha_8), \\
y_2 &= 50(1-\alpha_3)(1-\alpha_4)(1-\alpha_7) + 80(1-\alpha_5)(1-\alpha_8), \\
y_3 &= 50(1-\alpha_2)(1-\alpha_4)(1-\alpha_7), \\
y_4 &= 60(1-\alpha_1)(1-\alpha_7) + 50(1-\alpha_2)(1-\alpha_3)(1-\alpha_7), \\
y_5 &= 70(1-\alpha_1)(1-\alpha_8), \\
y_6 &= 80(1-\alpha_2)(1-\alpha_8), \\
y_7 &= 60(1-\alpha_1)(1-\alpha_4) + 50(1-\alpha_2)(1-\alpha_3)(1-\alpha_4), \\
y_8 &= 70(1-\alpha_1)(1-\alpha_5) + 80(1-\alpha_2)(1-\alpha_6).
\end{aligned}
$$

Numerical Procedure

In order to use the EFPM to determine the blocking probabilities in any network, the following numerical procedure can be invoked.

(1) Initially assume $\boldsymbol{\alpha} = (\alpha_j,\ j \in \mathcal{J}) = \mathbf{0}$.

(2) Calculate $\mathbf{y} = (y_j,\ j \in \mathcal{J})$ using the current value of $\boldsymbol{\alpha}$ and

$$y_j = \sum_{\{r:j\in r\}} a_r \prod_{\{i\in r, i\neq j\}} (1-\alpha_i).$$

(3) Calculate $\boldsymbol{\alpha}$ using the Erlang-B formula and $\boldsymbol{y}$ as calculated in Step 2.

(4) Compare the new estimate of $\boldsymbol{\alpha}$ with the previous estimate. If any component differs by more than some specified tolerance then go to Step 2; else

(5) Calculate $\mathbf{B} = (B_r, r \in \mathcal{R})$ according to $B_r = 1 - \prod_{j\in\mathcal{J}} (1-\alpha_j)^{A_{j,r}}$.
