Dimensionado y Planificación de Redes
Tema 4 – Modelo M/M/S/S y extensiones

Ramón Agüero Calvo
ramon.agueroc@unican.es

Contenido

Contenido

Motivación

- En un sistema de pérdida pura, cuando todos los recursos están ocupados las llamadas entrantes se pierden – no hay espera
- Su uso tradicional se restringía a las comunicaciones basadas en conmutación de circuitos
- El crecimiento de la tecnología IP y de la conmutación de paquetes hizo que se redujera la importancia de este tipo de modelos – auge de los sistemas de espera pura: M/M/1 y extensiones
- Las redes de nueva generación has supuesto un nuevo impulso a los modelos de pérdida pura
  - En la parte de acceso a través de las comunicaciones móviles: desde la 2G (GSM) hasta la 4G (LTE) es apropiado modelar el rendimiento en base a un sistema de pérdida pura
  - En la parte dorsal se tiende a un control de admisión (basado en criterios de QoS) que utiliza túneles virtuales, por lo que también se pueden emplear modelos de pérdida

Integración de servicios

- Complejidad añadida por la integración de servicios, cada uno de ellos con características diferentes
- La integración va más allá de los servicios en sí, y en las redes de nueva generación se habla del FMC (*Fixed Mobile Convergence*)
- Servicio *k* – $\{\lambda_k, \mu_k, b_k, PB_k\}$, con
  - λ_k – tasa de llegadas
  - μ_k – esperanza del tiempo de servicio (-1)
  - b_k – capacidad requerida (ancho de banda equivalente)
  - PB_k – requisito de QoS, probabilidad de bloqueo máxima

Contenido

Expresión original

- La fórmula de ErlangB *EB(S,A)* establece la probabilidad de bloqueo (o pérdida, con tráfico Poisson) cuando se ofrece un tráfico de A Erlangs a un conjunto de S recursos
- Se supone que las llegadas siguen un proceso de Poisson de tasa λ y que el tiempo de servicio se puede modelar con una variable aleatoria exponencial negativa, con tasa $\mu - A = \frac{\lambda}{\mu}$

$$PB = EB(S,A) = \frac{\frac{A^S}{S!}}{\sum_{j=0}^{S}\left(\frac{A^j}{j!}\right)}$$

Cálculo de Erlang-B recursivo

- Tiempo de cálculo elevado; complejidad de las operaciones (!)
- Utilización de la fórmula recursiva, con *EB(0,A) = 1*

$$EB(S,A) = \frac{A \cdot EB(S-1,A)}{S + A \cdot EB(S-1,A)}$$

- El tiempo de cálculo sigue siendo elevado
- No se puede despejar S ni A, por lo que es habitual emplear tablas o gráficas

Fórmula de Erlang-B - Tablas

Servidores necesarios para un tráfico ofrecido dado

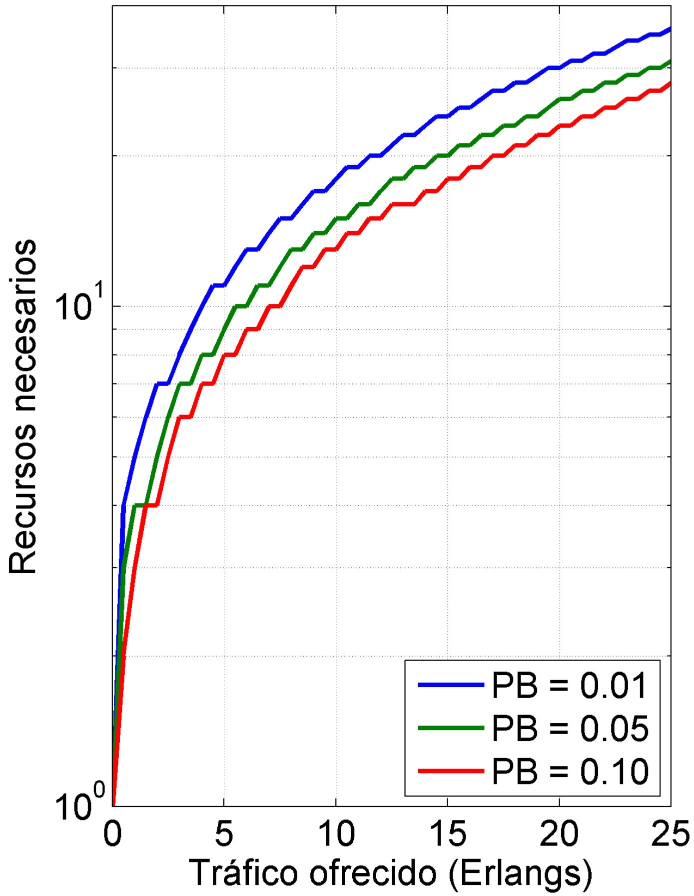
A (Erl)	Servidores		
	PB = 0.01	PB = 0.05	PB = 0.10
0.5	4	3	2
1.0	5	4	3
1.5	6	4	4
2.0	7	5	4
2.5	7	6	5
3.0	8	7	6
3.5	9	7	6
4.0	10	8	7
4.5	11	8	7
5.0	11	9	8
5.5	12	10	8
6.0	13	10	9
6.5	13	11	9
7.0	14	11	10
7.5	15	12	10
8.0	15	13	11
8.5	16	13	12
9.0	17	14	12
9.5	17	14	13
10.0	18	15	13

Tráfico máximo para un número de servidores dado

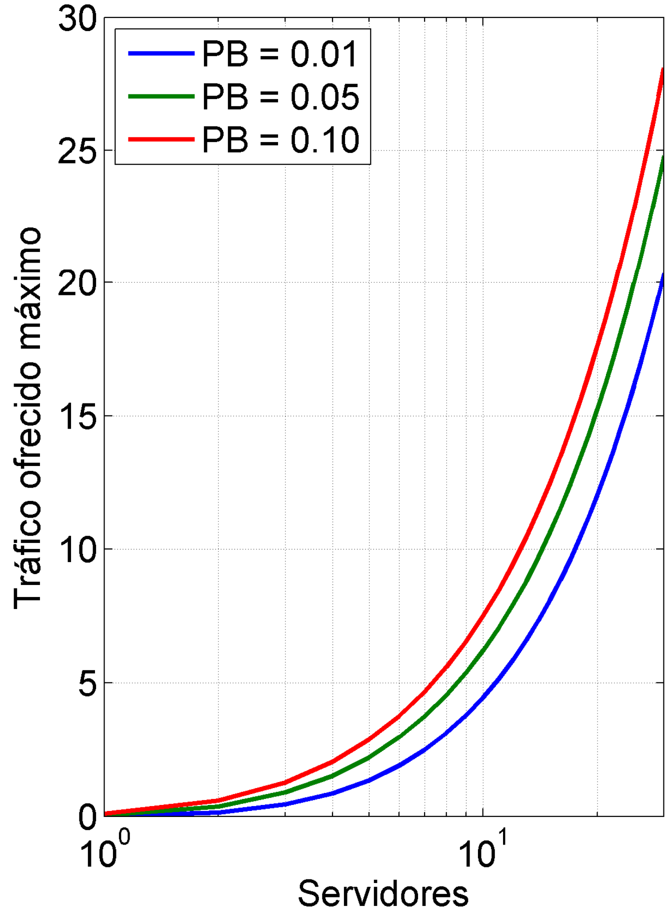
S	A (Erlangs)		
	PB = 0.01	PB = 0.05	PB = 0.10
1	0.000	0.041	0.102
2	0.143	0.367	0.592
3	0.449	0.898	1.265
4	0.857	1.510	2.041
5	1.347	2.204	2.878
6	1.898	2.959	3.755
7	2.490	3.735	4.653
8	3.122	4.531	5.592
9	3.776	5.367	6.531
10	4.449	6.204	7.510
11	5.143	7.061	8.469
12	5.857	7.939	9.469
13	6.592	8.816	10.469
14	7.347	9.714	11.469
15	8.102	10.633	12.469
16	8.857	11.531	13.490
17	9.633	12.449	14.510
18	10.429	13.367	15.531
19	11.224	14.306	16.571
20	12.020	15.245	17.612

Fórmula de Erlang-B - Gráficas

Servidores necesarios para un tráfico ofrecido dado



Tráfico máximo para un número de servidores dado



Cálculo de Erlang-B aproximado

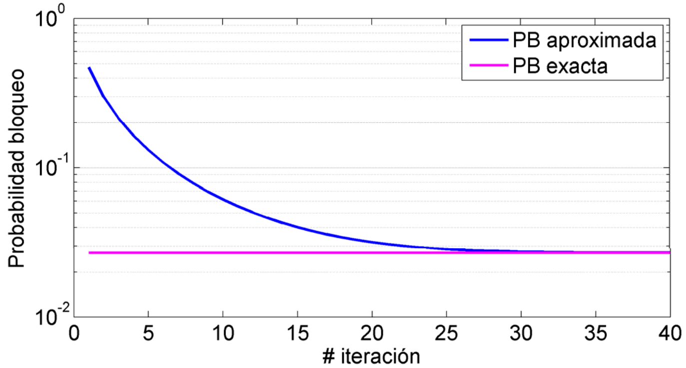
- Utilizando la siguiente expresión para la fórmula de Erlang-B

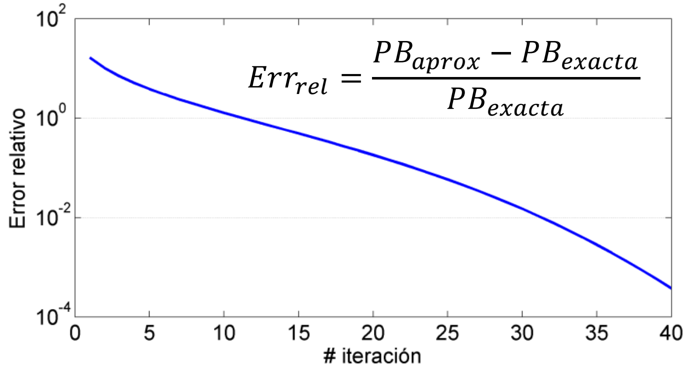
$$EB(S,A)^{-1} = \sum_{i=0}^{S} S \cdot (S-1) \cdot \ldots \cdot (S-i+1) \cdot A^{-i}$$

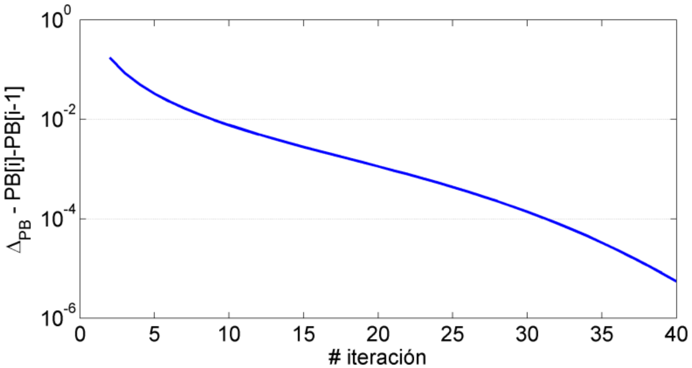
- Se pueden hacer iteraciones hasta que el valor converja (diferencia pequeña con la iteración previa)
  - Para valores de S > 100 y PB << 1 la convergencia es rápida y el cálculo se puede terminar pronto

Cálculo de Erlang-B aproximado

- Ejemplo con S = 100, A = 90 Erlangs
  - La probabilidad de bloqueo exacta es 0.0269574
  - Se puede ver que la convergencia es apropiada (para ≈30 iteraciones se consiguen valores cercanos al exacto)







Estimación de Jagerman

- Estimación(1) que permite trabajar con valores de S reales (x)

$$EB(x,A)^{-1} \approx a_0(c)\,\sqrt{x} + a_1(c) + \frac{a_2(c)}{\sqrt{x}}$$

siendo c la carga normalizada

$$c = \frac{A-x}{\sqrt{x}}$$

y a_0, a_1 y a_2 coeficientes que se pueden encontrar en forma de tabla

$$a_0(c) = e^{\frac{1}{2}c^2}\int_c^{\infty} e^{-\frac{1}{2}u^2}du \qquad a_1(c) = \frac{2}{3} + \frac{1}{3}c^2 - \frac{1}{3}c^3 a_0(c)$$

$$a_2(c) = -\frac{1}{18}c^5 - \frac{7}{36}c^3 + \frac{1}{12}c + \left(\frac{1}{18}c^6 + \frac{1}{4}c^4 + \frac{1}{12}\right)a_0(c)$$

(1) D. L. Jagerman "Some Properties of the Erlang Loss Function". The Bell System Technical Journal. Vol. 53, No 3 (Marzo 1974)

Características de la Fórmula de Erlang-B

- Si el número de recursos S crece (S → ∞) y el tráfico (A) se mantiene, la probabilidad de bloqueo tiende a 0: $\lim_{\substack{S\to\infty \\ A\ \mathrm{cte}}} EB(S,A) = 0$
- Cuando el tráfico también crece, pero se puede establecer el siguiente límite $\lim_{\substack{S\to\infty \\ A\to\infty}} \frac{A}{S} = \gamma$, se pueden dar dos supuestos diferentes

$$\lim_{\substack{S,A\to\infty \\ \frac{A}{S}\to\gamma}} EB(S,A) = \begin{cases} 0 & \gamma \le 1 \\ 1 - \dfrac{1}{\gamma} & \gamma > 1 \end{cases}$$

- Para A elevado y PB << 1 se tiene que…

$$PB \approx 1 - \frac{1}{\gamma} = 1 - \frac{S}{A} \to S \approx A \cdot (1 - PB)$$

al utilizar la expresión anterior para dimensionar sistemas se hace una sobre-estimación

Características de la Fórmula de Erlang-B

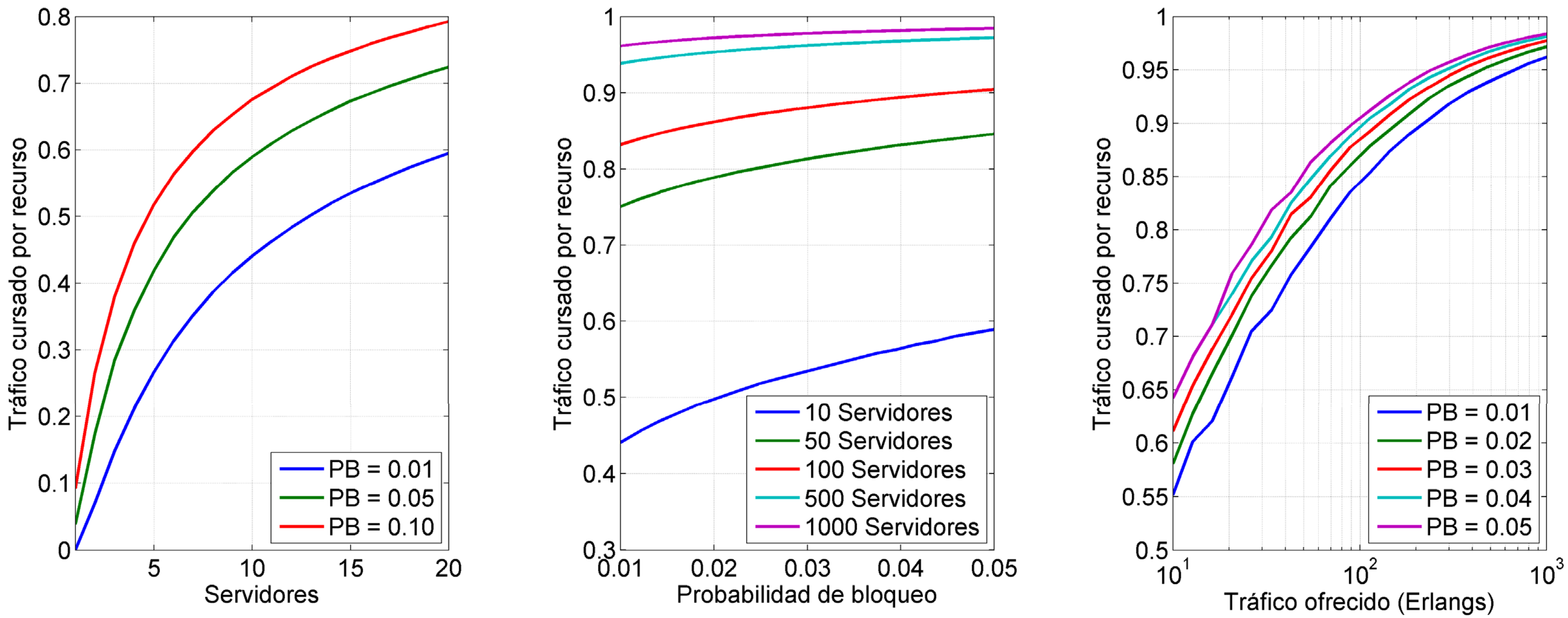
- Se ha demostrado que la fórmula de Erlang-B es válida para cualquier distribución del tiempo de servicio
- Además también es válida para cualquier política de ocupación de los servidores (*hunting*)
  - Secuencial
  - Secuencial desde el último ocupado
  - Aleatorio
- La ocupación individual de los recursos sí depende de la política de ocupación de los recursos

Ocupación (TC) por servidor

- Para una política de ocupación aleatoria, el tráfico cursado (ocupación) se reparte equitativamente entre todos los recursos

$$(TC)_{\text{recurso}} = \frac{(TC)_{\text{total}}}{S} = \frac{A \cdot (1 - PB)}{S} = \frac{A \cdot \left(1 - EB(S, A)\right)}{S}$$

  - El TC por recurso es su utilización, por lo que $(TC)_{\text{recurso}} < 1$



Contenido

Modelo

- Se asume que las llegadas que se rechazan vuelven a intentar acceder a un recurso (de manera inmediata) con probabilidad P_L, por lo que…

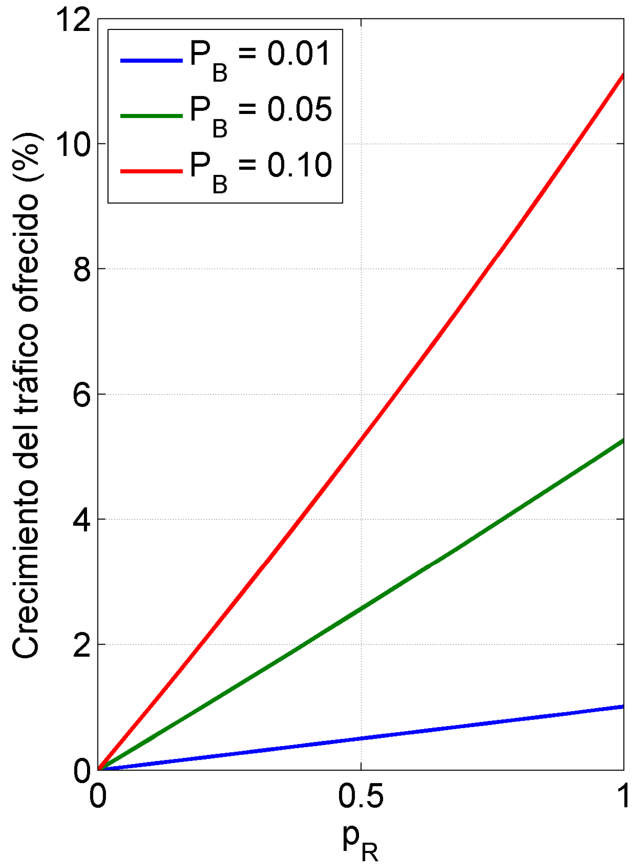
$$A_T = A + A_T \cdot PB \cdot P_L \rightarrow A_T = \frac{A}{1 - PB \cdot P_L}$$

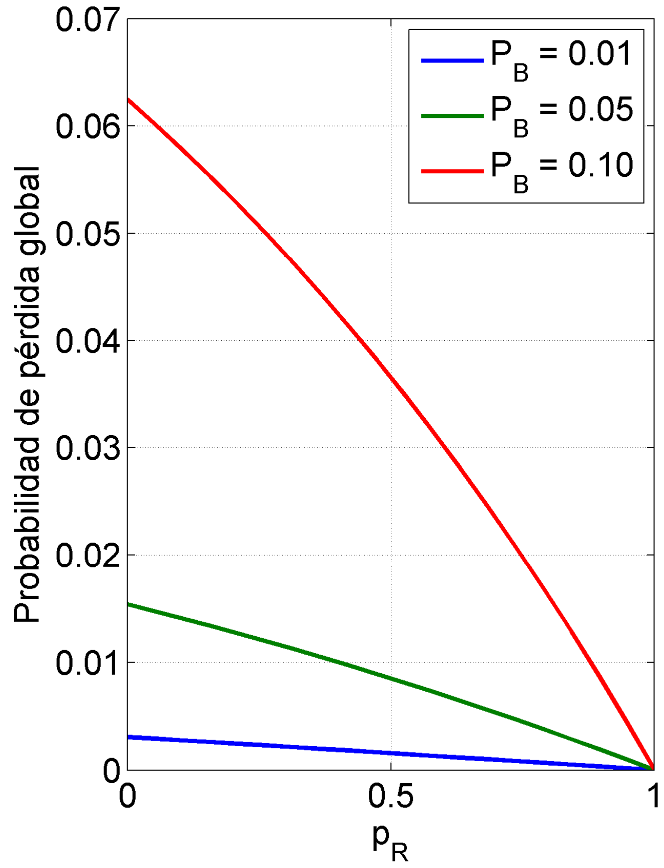
donde PB es la probabilidad de bloqueo del sistema (considerando todo el tráfico ofrecido) PB = EB(S, A_T)

- Limitaciones / simplificaciones
  - Se asume que las llamadas rechazadas pueden reintentarse de manera indefinida (no se limita el número máximo de reintentos)
  - Se asume que el tráfico generado por las llamadas que se reintentan es de Poisson, lo que no es cierto

Ejemplo

- Notar la diferencia entre la probabilidad de bloqueo del sistema y la probabilidad de bloqueo global, con A = 1 Erlang





Contenido

Introducción

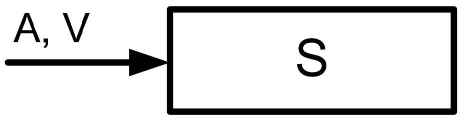
- La fórmula de Erlang-B es válida siempre que el proceso de llegada siga una distribución de Poisson
  - Ya se ha visto que la distribución del tiempo de servicio no limita su uso
- Se considera ahora un tráfico genérico, con media A y varianza V
- Se define el factor de *peakedness* o *burstiness* (índice de dispersión) como el cociente entre la varianza y el valor medio (*variance-to-mean ratio, VMR*)

$$VMR = \frac{V}{A}$$

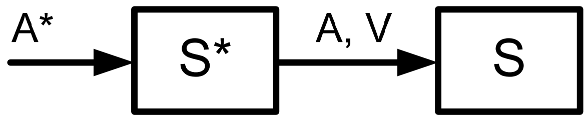
- En función del valor del VMR se pueden dar tres casos
  - VMR > 1 – tráfico a ráfagas (*peaked*)
  - VMR = 1 – tráfico aleatorio (distribución de Poisson)
  - VMR < 1 – tráfico suave (*smoothed*)

Aplicación: Método de Wilkinson[1]

- El modelo surge de un sistema con desbordamiento (ver siguiente sección), pero puede generalizarse para cualquier tráfico en el que VMR > 1
- Se utiliza el modelo que se muestra en la figura



Sistema original



Sistema equivalente

- Se elige el valor de A* de manera que el tráfico de entrada al sistema equivalente sea de Poisson
  - Se suele utilizar la aproximación de Rapp para calcular A* y S*

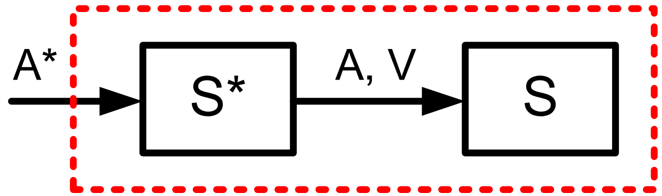
$$A^* = V + 3 \cdot VMR \cdot (VMR - 1)$$

$$S^* = \frac{A^*}{VMR - 1 + A} + A^* - A - 1$$

(1) R. I. Wilkinson "Theories for Toll Traffic Engineering in the USA". The Bell System Technical Journal. Vol. 35, No 2 (Marzo 1956)

Aplicación: Método de Wilkinson

- La pérdida total del sistema equivalente se puede calcular a través de la fórmula de Erlang-B: PB* = EB(S*+S,A*)



Sistema equivalente
- S+S recursos*
*- Tráfico de Poisson A**

- Además, se sabe que la pérdida del primer grupo (PB_S*) es EB(S*,A*), por lo que…

$$A\big(1-PB(S,A)\big) = A^* \cdot PB(S^*,A^*) - A^* \cdot PB(S+S^*,A^*) \rightarrow$$
$$\rightarrow A^* \cdot PB(S^*,A^*)\big(1-PB(S,A)\big) = A^* \cdot PB(S^*,A^*) - A^* \cdot PB(S+S^*,A^*)$$

llegando, finalmente, a…

$$PB(S,A) = \frac{PB(S+S^*,A^*)}{PB(S^*,A^*)}$$

- Como se puede ver el número de servidores “equivalente” puede ser real: uso de la expresión de Jagerman o interpolación

Aplicación: Método de Fredericks[1]

- Se calcula la probabilidad de bloqueo del sistema aplicando la fórmula de Erlang-B a un sistema "equivalente"

$$PB \approx EB\left(\frac{S}{VMR}, \frac{A}{VMR}\right)$$

  - Como se puede ver el número de servidores "equivalente" puede ser real: uso de la expresión de Jagerman o interpolación
- En función del tipo de tráfico la estimación será…
  - Si el tráfico es *a ráfagas* (VMR > 1) la PB que se obtiene es menor que la exacta
    - Si se hubiera utilizado la fórmula de Erlang-B con S y A, la PB resultante sería notablemente inferior
  - Si el tráfico es *suave* (VMR < 1), la PB será mayor que la exacta
    - Si se hubiera utilizado la fórmula de Erlang-B con S y A, la PB resultante sería aún mayor

(1) A. A. Fredericks "Congestion in Blocking Systems – A Simple Approximation Technique". The Bell System Technical Journal. Vol. 59, No 6 (Julio/Agosto 1980)

Contenido

Introducción

- El tráfico de desbordamiento es aquel que se ofrece a un segundo grupo de recursos cuando uno anterior está bloqueado
- Cuando se da esa situación, el proceso de llegadas es de Poisson, pero cuando la congestión desaparece, no se ofrece tráfico alguno al segundo grupo: *Interrupted Poisson Process*
- Este tipo de tráfico fue muy utilizado en las redes de conmutación de circuitos y recientemente ha vuelto a ganar relevancia con las redes de comunicaciones móviles – *células paraguas*
- Se pueden analizar aplicando el método de Erlang-B extendido (Wilkinson o Fredericks), utilizando el valor medio y la varianza del tráfico desbordado

Características del tráfico desbordado

- Se pueden emplear las fórmulas propuestas originalmente por Kosten

$$E\left(A^d\right) = A^d = A \cdot PB \qquad V\left(A^d\right) = A^d\left[1 - A^d + \frac{A}{1 + S - A + A^d}\right]$$

  donde A es el tráfico ofrecido al primer grupo de circuitos (S), y PB es la probabilidad de bloqueo correspondiente: PB = EB(S, A)

- Se puede demostrar que el VMR para el tráfico de desbordamiento es mayor de la unidad, por lo que se trata de un tráfico *peaked* (a ráfagas)
- Es posible que al tráfico desbordado se le "añada" otra corriente de tráfico, modificando su valor medio y varianza

Desbordamiento – Ejemplo 1

- Se pretende que PB_{final} = 1% y se asume que S_1 = A
  - En este caso $PB_2 = PB_{final}/PB_1$

1) Asumiendo tráfico Poisson
2) Fredericks con servidores "reales" Jagerman
3) Fredericks con servidores "enteros"
4) Sin desbordamiento: A se ofrece a un único grupo

A	$E(A_d)$	$V(A_d)$	VMR	$S_1 + S_2$			
				(1)	(2)	(3)	(4)
10	2.146	4.36	2.033	15	18	19	18
20	3.178	8.29	2.609	26	30	31	30
30	3.973	12.15	3.058	37	41	43	42
40	4.646	15.97	3.438	48	52	54	53
50	5.239	19.77	3.774	58	63	66	64
60	5.776	23.56	4.079	69	74	77	75
70	6.270	27.33	4.359	79	84	88	85
80	6.729	31.09	4.621	89	95	99	96
90	7.161	34.85	4.866	100	106	110	107
100	7.570	38.60	5.098	110	116	116	117
200	10.870	75.86	6.978	212	220	221	221
300	13.404	112.90	8.422	313	322	326	324
400	15.541	149.83	9.641	414	423	429	426
500	17.424	186.69	10.715	515	524	533	527
600	19.126	223.51	11.686	615	625	636	628
700	20.692	260.28	12.579	716	726	738	728
800	22.149	297.02	13.410	816	826	827	829
900	23.517	333.74	14.191	916	926	929	929
1000	24.812	370.44	14.930	1017	1026	1030	1029

Desbordamiento – Ejemplo 2

- Al segundo grupo se añade un tráfico directo (Poisson) $A_2 = 0.5 \cdot A_1$
  - En este caso $PB_2 = PB_{final}$

1) Asumiendo tráfico Poisson
2) Fredericks con servidores "reales" Jagerman
3) Fredericks con servidores "enteros"
4) Sin desbordamiento: A se ofrece a un único grupo

A	$E(A_2)$	$V(A_2)$	VMR	$S_1 + S_2$			
				(1)	(2)	(3)	(4)
10	7.15	9.36	1.310	24	26	26	24
20	13.18	18.29	1.388	42	44	44	42
30	18.97	27.15	1.431	59	62	62	58
40	24.65	35.97	1.460	76	79	78	75
50	30.24	44.77	1.481	92	95	96	91
60	35.78	53.56	1.497	108	112	113	107
70	41.27	62.33	1.510	124	128	129	123
80	46.73	71.09	1.521	140	144	144	138
90	52.16	79.85	1.531	156	161	161	154
100	57.57	88.60	1.539	172	177	177	170
200	110.87	175.85	1.586	329	335	335	324
300	163.40	262.90	1.609	484	491	492	476
400	215.54	349.83	1.623	637	646	647	628
500	267.42	436.69	1.633	791	800	801	779
600	319.13	523.51	1.640	943	954	955	929
700	370.69	610.28	1.646	1096	1107	1107	1079
800	422.15	697.02	1.651	1248	1260	1261	1230
900	473.52	783.74	1.655	1400	1412	1414	1379
1000	524.81	870.44	1.659	1552	1565	1566	1529

Contenido

- Introducción
- Fórmula de Erlang-B
- Sistemas con reintento
- Métodos Erlang-B extendidos
- Tráfico de desbordamiento
- Sistemas de pérdida con múltiples servicios

Introducción

- Se considera un sistema con una capacidad finita C al que se le ofrece diferentes flujos de tráfico
  - $\{\lambda_j, \mu_j, b_j\}$ $j = 1\ldots K$, con

    λ_j: tasa de llegadas del servicio j-ésimo (proceso de Poisson)

    μ_j: inverso del tiempo medio para el servicio j-ésimo (va exponencial negativa)

    b_j: capacidad requerida por el servicio j-ésimo
- Las peticiones que llegan se aceptan si el sistema tiene capacidad suficiente, en caso contrario se pierden: *Sistema de pérdida pura*
- No se asignan prioridades a ningún tipo de servicio (por ejemplo, reservando capacidades)
- Asignar recursos de un *pool* común a medida que van llegando peticiones: *Stochastic Knapsack Model* – Modelo de la Mochila Estocástica

Modelo

- Espacio de estados $\mathcal{S} \coloneqq \left\{\boldsymbol{n} \in \mathcal{I}^{\mathcal{K}}: \boldsymbol{b} \cdot \boldsymbol{n} \leq C\right\}$, con
  - $\boldsymbol{n} = \{n_1, n_2 \dots n_K\}$: vector que indica el número de unidades de cada servicio en el sistema
  - $\boldsymbol{b} = \{b_1, b_2 \dots b_K\}$: vector que indica la capacidad requerida por cada tipo de servicio
  - $\mathcal{I}^{\mathcal{K}}$: producto cartesiano (K dimensiones) de los números naturales
  - Notar, además, que $\boldsymbol{b} \cdot \boldsymbol{n} = \sum_{j=1}^{K} b_j n_j$
- Se asume que el sistema tiene equilibrio, y se define $\pi(\boldsymbol{n})$ como la probabilidad de que el sistema se encuentre en el estado ***n***

$$\pi(\boldsymbol{n}) = \frac{1}{G} \prod_{j=1}^{K} \frac{\rho_j^{n_j}}{n_j!} \qquad\qquad G = \sum_{\boldsymbol{n} \in \mathcal{S}} \prod_{j=1}^{K} \frac{\rho_j^{n_j}}{n_j!}$$

con $\rho_j = \frac{\lambda_j}{\mu_j}$, tráfico ofrecido por el servicio j-ésimo

Bloqueo

- Se define $\mathcal{S}_k$ como el subconjunto de estados en los que se admitirían una llegada del servicio *k*

$$\mathcal{S}_k := \{n \in \mathcal{S} : b \cdot n \leq C - b_k\}$$

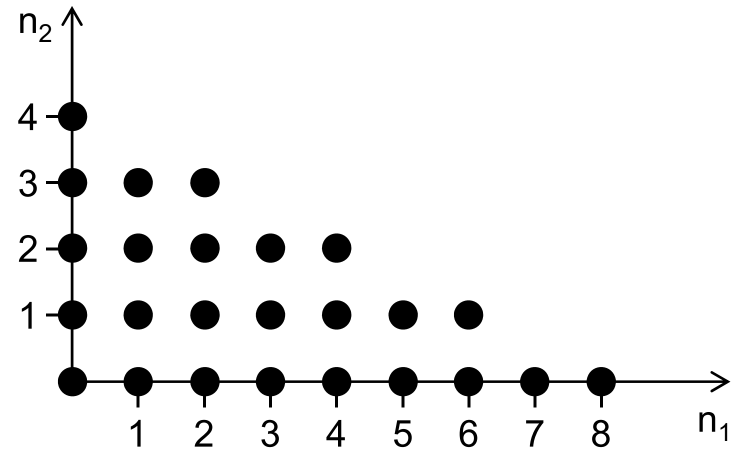
- La probabilidad de bloqueo para un servicio de tipo *k*, será por tanto…

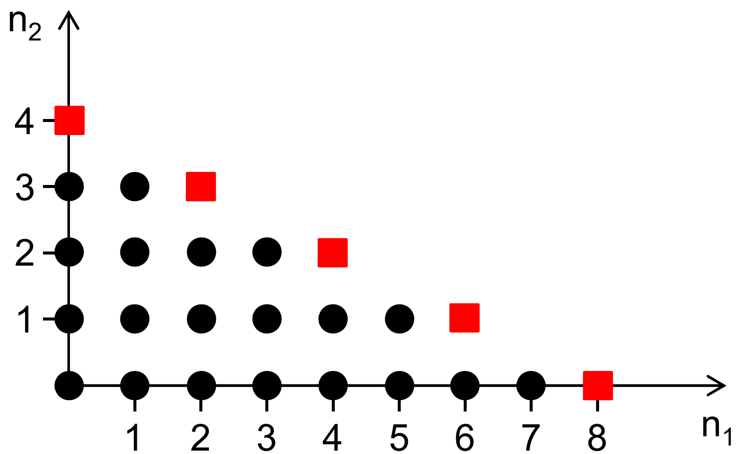
$$B_k = 1 - \sum_{n \in \mathcal{S}_k} \pi(n) = 1 - \frac{\sum_{n \in \mathcal{S}_k} \prod_{j=1}^{K} \frac{{\rho_j}^{n_j}}{n_j!}}{\sum_{n \in \mathcal{S}} \prod_{j=1}^{K} \frac{{\rho_j}^{n_j}}{n_j!}}$$

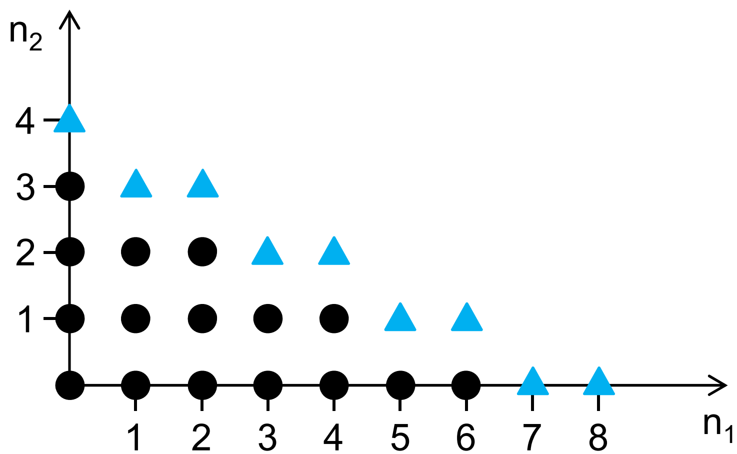
- El resultado anterior es poco práctico, debido al tamaño de los conjuntos correspondientes, lo que hace el cálculo muy caro computacionalmente

Bloqueo – Ejemplo

- Capacidad C = 8; dos servicios, con capacidades $b_1 = 1$ y $b_2 = 2$
- El conjunto S es el que se muestra en la figura de la derecha
- Las figuras de abajo representan
  - Conjunto $S\text{-}S_1$ (izquierda) ■
  - Conjunto $S\text{-}S_2$ (derecha) ▲







Bloqueo – Algoritmo recursivo(1)

- Manera eficiente de calcular las probabilidades de bloqueo de manera recursiva
- Definiciones previas
  - Conjunto de estados de capacidad c: $\mathcal{S}(c) \coloneqq \{\boldsymbol{n} \in \mathcal{S} \ : \ \boldsymbol{b} \cdot \boldsymbol{n} = c\}$
  - Probabilidad de que la capacidad ocupada del sistema sea c: $q(c) := \sum_{\boldsymbol{n} \in \mathcal{S}(c)} \pi(\boldsymbol{n})$
  - Número medio de servicios de clase k cuando la capacidad ocupada del sistema es c: $R_j(c) \coloneqq \sum_{\boldsymbol{n} \in \mathcal{S}(c)} n_j \pi(\boldsymbol{n})$
- Se puede demostrar que se cumple la siguiente ecuación recursiva

$$cq(c) = \sum_{j=1}^{K} b_j \rho_j q(c - b_j)$$

(1) J. S. Kaufman “Blocking in a shared resource environment”. IEEE Transactions of Communications. Vol. 29, No 10 (1981)

Bloqueo – Algoritmo recursivo

(1) g(0) = 1; g(c) = 0 if c < 0

(2) for c = 1…C

$$g(c)=\frac{1}{c}\sum_{j=1}^{K} b_j \rho_j g(c-b_j)$$

(3) $G=\sum_{c=0}^{C} g(c)$

(4) for c = 0…C

$$q(c)=\frac{g(c)}{G}$$

(5) for j = 1…K

$$PB_j=\sum_{c=C-b_j+1}^{C} q(c)$$

Aproximación alternativa

- Para evitar el uso del algoritmo anterior se pueden normalizar las capacidades y utilizar un tráfico equivalente para obtener una probabilidad de bloqueo única
- Si se utiliza el servicio *r* para normalizar, se tiene que…

$$A_e = \sum_{j=1}^{K} \rho_j \left(\frac{b_j}{b_r}\right) \qquad V(A_e) = \sum_{j=1}^{K} \rho_j \left(\frac{b_j}{b_r}\right)^2 \qquad VMR = \frac{\sum_{j=1}^{K} \rho_j \left(\frac{b_j}{b_r}\right)^2}{\sum_{j=1}^{K} \rho_j \left(\frac{b_j}{b_r}\right)}$$

- Aplicando el método de Fredericks, se obtienen los recursos *x* necesarios para garantizar la probabilidad de bloqueo (se utiliza la más restrictiva)

$$x : \; EB\left(x, \frac{A_e}{VMR}\right) < PB_{min}$$

- Finalmente, la capacidad final resulta $C = x \cdot VMR \cdot b_r$

Ejemplo (1)

- Se consideran 100 usuarios que pueden generar cuatro tipos de servicios (ver tabla) y se pretende dimensionar la capacidad, para que la PB sea 1%

	VoIP (1)	Streaming (2)	Datos (3)	Best Effort (4)
BW (kbps)	100	800	400	200
α (h^{-1}) - BH	2	0.2	0.1	0.3
Ts (m)	3	12	15	30
λ (h^{-1}) – BH	200	20	10	30
ρ - BH	10	4	2.5	15

$$(A_e)_{r=1} = \sum_{j=1}^{K} \rho_j \left(\frac{b_j}{b_1}\right) = 82\ E \qquad V(A_e)_{r=1} = \sum_{j=1}^{K} \rho_j \left(\frac{b_j}{b_1}\right)^2 = 366\ E^2$$

$$(A_e)_{r=2} = \sum_{j=1}^{K} \rho_j \left(\frac{b_j}{b_2}\right) \approx 10.25\ E \qquad V(A_e)_{r=2} = \sum_{j=1}^{K} \rho_j \left(\frac{b_j}{b_2}\right)^2 \approx 5.72\ E^2$$

Ejemplo (2)

- Para las dos normalizaciones, el cociente Ae/VMR es igual (≈ 18.37)
- Aplicando el método de Fredericks y la expresión de Jagerman, se obtiene, en ambos casos que…

$$x:\ PB = EB\left(x, \frac{A_e}{VMR}\right) \leq 0.01 \xrightarrow{\text{Jagerman}} x \geq 27.85$$

- Con lo que se puede estimar la capacidad necesaria en el sistema…

$$C = x \cdot VMR \cdot b_r = \begin{cases} 27.85 \cdot 4.46 \cdot 100 \approx 12430\ kbps \\ 27.85 \cdot 0.56 \cdot 800 \approx 12430\ kbps \end{cases}$$

- Se puede ver que el servicio empleado para la normalización no afecta al resultado final

Ejemplo (3)

- Aplicando el algoritmo recursivo se puede obtener la probabilidad de bloqueo exacta para cada tipo de servicio
- Se asume que b = {1,8,4,2} y que C = 128 (algo mayor que la obtenida con el método aproximado)
- Se obtiene que PB ≈ {**0.0015**, **0.0173**, **0.0071**, **0.0032**}