Single Server Retrial Queueing System with Orbital Search under Erlang-K Service

Gopal Sekar
Associate Professor, Tagore Arts College, Pondicherry
Email: gopsek28@yahoo.co.in

G.Ayyappan
Associate Professor, Pondicherry Engineering College, Pondicherry
Email: ayyappanpec@hotmail.com

A. Muthu Ganapathi Subramanian
Associate Professor, Tagore Arts College, Pondicherry
Email: csamgs1964@gmail.com

ABSTRACT

Consider a single server retrial queueing system with orbital search in which customers arrive in a Poisson process with arrival rate λ. Let k be the number of phases in the service station. Let the service time follow an Erlang-k distribution with service rate $k\mu$ for each phase. The concept of orbital search is introduced in this paper. We assume that the services in all phases are independent and identical and only one customer at a time is in the service mechanism. If the server is **free** at the time of a primary call arrival, the arriving call begins to be served in Phase 1 immediately by the server then progresses through the remaining phases and must complete the last phase and leave the system before the next customer enters the first phase. If the server is **busy**, then the arriving customer goes to orbit and becomes a source of **repeated calls**. This pool of sources of repeated calls may be viewed as a sort of queue. Every such source produces a Poisson process of repeated calls with intensity σ. If an incoming repeated call finds the server free, it is served in the same manner and leaves the system after service, while the source which produced this repeated call disappears. After completing each service, with probability p, the server searches a customer from the orbit and starts the service and with probability (1-p) continues to be idle in the system. We assume that the access from orbit to the service facility is governed by the constant retrial policy. This model is solved using **Direct Truncation Method**. Numerical study have been done for Analysis of Mean number of customers in the orbit (MNCO),Truncation level (OCUT),Probabilities of server free and busy for various values of λ, μ, k, p and σ in elaborate manner and also various particular cases of this model have been discussed.

Keywords : Single Server – Erlang-k distribution – Direct Truncation Method – classical retrial policy – Orbital search

9. H.Uesu : Fuzzy Node Fuzzy Graph, T-Norm and its Application,Journal of Japan Society for Fuzzy Theory andIntelligent Informatics Vol.16 No.1(2004)88–95, Japanese

10. H.Uesu, H.Yamashita, T.Takizawa and M.Yanai : OptimalFuzzy Graph Based on Fuzzy Node Fuzzy GraphAnalysis, The 3rd International Conference on Soft Computing and Intelligent Systems(2006)

11. H.Uesu, S.Kimiaki and H.Yamashita : Analysis ofFuzzy Node Fuzzy Graph and its Application, Int'l Conferenceon Soft Computing and Human Science(2007).

1. INTRODUCTION

The main objective of this research paper is to analyze the behaviour of Retrial queueing systems with Orbital search under Erlang-k service. Queueing systems in which arriving customers who find all servers and waiting positions (if any) occupied may retry for service after a period of time are called **Retrial queues**. Models of retrial queues are an important part of queueing theory. These models arise because of the necessity to allow for the retrial effect in various networking systems and in day to day life. Therefore, much attention is paid to the analysis of such models of queues. Retrial queueing models accurately describe the operation of many telecommunication networks. So their investigation is very important. The detailed survey of retrial queues and bibliographical information have been obtained from Artalejo [1, 2, 3, 4], Falin [12], monograph by Falin and Templeton [13], Yang and Templeton [15], Ayyappan *et al.* [5, 6, 7, 8, 9, 10, 11] have studied the Single server retrial queueing under Erlang- service for various behaviour of Server and customer.

2. MODEL DESCRIPTION

Consider a single server retrial queueing system with orbital search [14] in which customers arrives in a Poisson process with arrival rate λ. Let k be the number of phases in the service station. Let the service time follow an Erlang-k distribution with service rate $k\mu$ for each phase. The concept of orbital search is introduced in this paper. We assume that the services in all phases are independent and identical and only one customer at a time is in the service mechanism. If the server is **free** at the time of a primary call arrival, the arriving call begins to be served in Phase 1 immediately by the server then progresses through the remaining phases and must complete the last phase and leave the system before the next customer enters the first phase. If the server is **busy**, then the arriving customer goes to orbit and becomes a source of **repeated calls**. This pool of sources of repeated calls may be viewed as a sort of queue. Every such source produces a Poisson process of repeated calls with intensity σ. If an incoming repeated call finds the server free, it is served in the same manner and leaves the system after service, while the source which produced this repeated call disappears. After completing each service, with probability p, the server searches a customer from the orbit and starts the service and with probability (1-p) continues to be idle in the system.

2.1. RETRIAL POLICY

We assume that the access from the orbit to the service facility follows an exponential distribution with rate $n\sigma$ which may depend on the current number n, $(n \geq 0)$ the number of customers in the orbit. That is, the probability of repeated attempt during the interval $(t, t+\Delta t)$, given that there are n customers in the orbit at time t is nσ"t. It is called the classical retrial policy. The input flow of primary calls, interval between repetitions and service times in phases are mutually independent.

2.3. RANDOM PROCESS

Let N(t) be the random variable which represents the number of customers in the orbit at time t and S(t) represents the status of the server at time t.

The random process is described as

$\{<N(t), S(t))> / N(t) = 0, 1, 2, \ldots; S(t) = 0, 1, 2, 3, \ldots, k\}$

The possible state spaces is

$\{ (u, v) / u = 0, 1, 2, 3, \ldots; v = 0, 1, 2, 3, \ldots, k\}$

The infinitesimal generator matrix **Q** is given below

$$Q = \begin{pmatrix} A_{00} & A_0 & O & O & O & \ldots \\ A_{10} & A_{11} & A_0 & O & O & \ldots \\ O & A_{21} & A_{22} & A_0 & O & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \end{pmatrix}$$

The matrices described in the Infinitesimal generator matrix Q can be obtained from the following infinitesimal transition rates of process X as follows.

For i = 0

$$q_{(i,j)(l,m)} = \begin{cases} \lambda & if\ (l,m) = (i, j+1)\ for\ j = 0 \\ -\lambda & if\ (l,m) = (i, j)\ for\ j = 0 \\ \lambda & if\ (l,m) = (i+1, j)\ for\ 1 \le j \le k \\ k\mu & if\ (l,m) = (i, j+1)\ for\ 1 \le j \le k-1 \\ k\mu & if\ (l,m) = (i, 0)\ for\ j = k \\ -(\lambda + k\mu) & if\ (l,m) = (i, j)\ for\ 1 \le j \le k \end{cases}$$

For i > 0

$$q_{(i,j)(l,m)} = \begin{cases} \lambda & if\ (l,m)=(i,j+1) & for\ j=0 \\ i\sigma & if\ (l,m)=(i-1,j+1) & for\ j=0 \\ -(\lambda+i\sigma) & if\ (l,m)=(i,j) & for\ j=0 \\ \lambda & if\ (l,m)=(i+1,j) & for\ 1\le j\le k \\ k\mu & if\ (l,m)=(i,j+1) & for\ 1\le j\le k-1 \\ pk\mu & if\ (l,m)=(i-1,1) & for\ j=k \\ (1-p)k\mu & if\ (l,m)=(i,0) & for\ j=k \\ -(\lambda+k\mu) & if\ (l,m)=(i,j) & for\ 1\le j\le k \end{cases}$$

3. DESCRIPTION OF COMPUTATIONAL METHOD

Retrial queueing models can be solved computationally by the following techniques.

(a) Direct Truncation Method

(b) Generalized Truncation Method

(c) Truncation Method using Level Dependent Quasi Birth- and –Death Process (LDQBD)

(d) Matrix Geometric Approximation

3.1 Direct Truncation Method

Let X be the steady-state probability vector of Q, partitioned as X = (x(0), x(1), x(2), . . .) where X satisfies

$$XQ = 0 \quad \text{and} \quad Xe = 1$$

where $x(i) = (P_{i0}, P_{i1}, P_{i2}, P_{i3}, \ldots, P_{ik})$; i = 0, 1, 2, . . .

The above system of equations can be solved by means of truncating the system of equations for sufficiently large value of the number of customers in the orbit, say M. That is, the orbit size is restricted to M such that any arriving customer finding the orbit full is considered lost. The value of M can be chosen so that the loss probability is small. Due to the intrinsic nature of the system, the only choice available for studying M is through algorithmic methods. While a number of approaches are available for determining the cut-off point, M, the one that seems to perform well is to increase M until the largest individual change in the elements of X for successive values is less than ε a predetermined infinitesimal value.

If M denotes the cut-off point or Truncation level, then the steady state probability vector $X^{(M)}$ is partitioned as $X^{(M)} = (x(0), x(1), x(2), \ldots, x(M))$, where $X^{(M)}$ satisfies

$$X^{(M)} Q^{(M)} = 0 \text{ and } X^{(M)} e = 1$$

where $x(i) = (P_{i0}, P_{i1}, P_{i2}, P_{i3}, \ldots, P_{ik})$; $i = 0, 1, 2, \ldots, M$

The above system of equations is solved exploiting the special structure of the co-efficient matrix. It is solved using Numerical methods. Since there is no clear cut choice for M, we may start the iterative process by taking, say M = 1 and increase it until the individual elements of X do not change significantly. That is, if M^* denotes the truncation point then

$\| x^{M^*}(i) - x^{M^*-1}(i) \|_\infty \le \varepsilon$ where ε is an infinitesimal quantity.

4. STABILITY CONDITION

Theorem: The necessary and sufficient condition for system to be stable is $\left(\frac{\lambda}{\mu}\right) < 1$

5. SPECIAL CASE

a. This model becomes a Single server retrial queueing system with Erlang-k service if $p \to 0$ which is discussed by Ayyappan *et al* [5].

b. This model becomes a standard queueing system if $p \to 1$

6. SYSTEM PERFORMANCE MEASURES

In this section we study some important performance measures along with their formula. These measures are used to bring out the qualitative behaviour of the queueing model under study. Numerical study has been dealt in very large scale to study the following measures. We can find various probabilities for various values of λ, μ, σ, p and k. The following system measures can be study with these probabilities.

a. The probability mass function of Server state

Prob (the server is idle) $= \sum_{i=0}^{\infty} p(i, 0)$

Prob (the server busy with a customer in th j^{th} phase) $= \sum_{i=0}^{\infty} p(i, j)$

b. The probability mass function of number of customers in the orbit

$$\text{Prob (no customers in the orbit)} = \sum_{j=0}^{k} p(0,j)$$

$$\text{Prob (i customers in the orbit)} = \sum_{j=0}^{k} p(i,j)$$

c. The Mean Number of Customers in the Orbit

$$\text{MNCO} = \sum_{i=0}^{\infty} i(\sum_{j=0}^{k} p(i,j))$$

d. The probability that the orbiting customer is blocked

$$\text{Blocking Probability} = \sum_{i=1}^{\infty}\sum_{j=1}^{k} p(i,j)$$

e. The probahility that an arriving customer enter into the service station immediately

$$= \sum_{i=0}^{\infty} p(i,0)$$

7. NUMERICAL STUDY

The values of parameters λ and μ are chosen so that they satisfy the stability condition discussed in section 4. The system performance measures of this model have been done and expressed in the form of tables which are shown below by finding the steady state probability vector X for various values of λ, μ, σ, p and k.

For $\lambda = 10$, $\mu = 20$, $\sigma = 100$, $p = 0.5$ and $k = 5$, the steady state probability vector is $X = (x(0), x(1), x(2), \ldots, x(M))$, where

X[0] = [0.1852, 0.0536, 0.0462, 0.0399, 0.0344, 0.0296]

X[1] = [0.0078, 0.0317, 0.0337, 0.0346, 0.0345, 0.0339]

X[2] = [0.0033, 0.0220, 0.0236, 0.0251, 0.0264, 0.0275]

X[3] = [0.0016, 0.0156, 0.0167, 0.0179, 0.0191, 0.0202]

X[4] = [0.0009, 0.0110, 0.0118, 0.0127, 0.0135, 0.0145]

X[5] = [0.0005, 0.0078, 0.0083, 0.0089, 0.0096, 0.0102]

X[6] = [0.0003, 0.0055, 0.0059, 0.0063, 0.0067, 0.0072]

X[7] = [0.0002, 0.0038, 0.0041, 0.0044, 0.0047, 0.0051]

X[8] = [0.0001, 0.0027, 0.0029, 0.0031, 0.0033, 0.0036]

X[9] = [0.0001, 0.0019, 0.0020, 0.0022, 0.0023, 0.0025]

X[10] = [0.0000, 0.0013, 0.0014, 0.0015, 0.0016, 0.0018]

Similarly, we can find x (n) for $n \geq 8$ and it is noticed that $x(n) \to 0$ as $n \to \infty$. For the numerical parameters chosen above, $x(n) \to 0$ for $n \geq 20$ and the sum of the steady state probabilities becomes 0.99999999999. In the same manner, we can find steady state probability vector X for all values of λ, μ, σ, p and k.

System Performance Measures

1. Probability that the server is idle = 0.200119
2. Probability that the server is busy with a customer in phase 1 = 0.159983

   Probability that the server is busy with a customer in phase 2 = 0.159975

   Probability that the server is busy with a customer in phase 3 = 0.159974

   Probability that the server is busy with a customer in phase 4 = 0.159965

   Probability that the server is busy with a customer in phase 5 = 0.159983
3. Probability that the server is busy = 0.799881
4. Probability mass function of number of customers in the orbit

No. of customers in the orbit	Probability	No. of customers in the orbit	Probability
0	0.388927	7	0.022387
1	0.176187	8	0.015702
2	0.128028	9	0.011003
3	0.091121	10	0.007706
4	0.064385	11	0.005393
5	0.045352	12	0.003773
6	0.031886	13	0.002638

1. Mean number of customers in the orbit = 2.064582
2. Probability that the orbiting customer is blocked = 0.596127

Notations:

OCUT : Truncation level (Orbit cut)

MNCO : Mean number of customers in the Orbit

P_0 : Probability that the server is idle

P_1 : Probability that the server is busy

Tables 1, 2 and 3 shows the impact of λ and retrial rate σ over Mean number of customers in the orbit and we infer the following

- Mean number of customers in the orbit decreases as retrial rate σ increases.
- Mean number of customers in the orbit increases as λ increases.
- P_0 and P_1 are independent of retrial rate σ.

Table 4 shows the effect of number of phases over the mean number of customers in the orbit and we infer the following

- Mean number of customers in the orbit decreases as number of phases increases

Table 5 shows the effect of p over the mean number of customers in the orbit and we infer the following

- As p tends to zero , it becomes Single server Retrial queueing system under Erlang-k service
- As p tends to 1, it becomes Standard Queueing system under Erlang-k service.

Table 1 System performance measures for $\lambda = 2$, $\mu = 10$, $p = 0.5$, $k = 5$ and various Values of σ

σ	OCUT		MNCO	P_0	P_1
10	8		0.0529	0.8000	0.2000
20	7		0.0419	0.8000	0.2000
30	7		0.0381	0.8000	0.2000
40	7		0.0361	0.8000	0.2000
50	7		0.0349	0.8000	0.2000
60	7		0.0341	0.8000	0.2000
70	7		0.0335	0.8000	0.2000
80	7		0.0331	0.8000	0.2000
90	7		0.0327	0.8000	0.2000
100	7		0.0325	0.8000	0.2000
200	7		0.0312	0.8000	0.2000
300	7		0.0308	0.8000	0.2000
400	7		0.0306	0.8000	0.2000
500	7		0.0305	0.8000	0.2000
600	7		0.0304	0.8000	0.2000
700	7		0.0304	0.8000	0.2000
800	7		0.0303	0.8000	0.2000
900	7		0.0303	0.8000	0.2000
1000	7		0.0302	0.8000	0.2000
2000	7		0.0301	0.8000	0.2000
3000	7		0.0301	0.8000	0.2000
4000	7		0.0301	0.8000	0.2000
5000	7		0.0300	0.8000	0.2000
6000	7		0.0300	0.8000	0.2000

Table 2 System performance measures for $\lambda = 5$, $\mu = 10$, $p = 0.5$, $k = 5$ and various Values of σ

σ	OCUT		MNCO	P_0	P_1
10	17		0.5129	0.5000	0.5000
20	16		0.4145	0.5000	0.5000
30	16		0.3784	0.5000	0.5000
40	16		0.3597	0.5000	0.5000
50	16		0.3482	0.5000	0.5000
60	16		0.3404	0.5000	0.5000
70	16		0.3348	0.5000	0.5000
80	16		0.3305	0.5000	0.5000
90	16		0.3272	0.5000	0.5000
100	16		0.3245	0.5000	0.5000
200	16		0.3124	0.5000	0.5000
300	16		0.3083	0.5000	0.5000
400	15		0.3062	0.5000	0.5000
500	15		0.3050	0.5000	0.5000
600	15		0.3042	0.5000	0.5000
700	15		0.3036	0.5000	0.5000
800	15		0.3031	0.5000	0.5000
900	15		0.3028	0.5000	0.5000
1000	15		0.3025	0.5000	0.5000
2000	15		0.3012	0.5000	0.5000
3000	15		0.3008	0.5000	0.5000
4000	15		0.3006	0.5000	0.5000
5000	15		0.3005	0.5000	0.5000
6000	15		0.3004	0.5000	0.5000
9000	15		0.3003	0.5000	0.5000

Table 3 System performance measures for $\lambda = 8$, $\mu = 10$, $p = 0.5$, $k = 5$ and various Values of σ

σ	OCUT	MNCO	P_0	P_1
10	51	3.3298	0.2000	0.8000
20	49	2.6605	0.2000	0.8000
30	48	2.4244	0.2000	0.8000
40	47	2.3030	0.2000	0.8000
50	47	2.2288	0.2000	0.8000
60	47	2.1787	0.2000	0.8000
70	47	2.1426	0.2000	0.8000
80	47	2.1154	0.2000	0.8000
90	46	2.0941	0.2000	0.8000
100	46	2.0770	0.2000	0.8000
200	46	1.9992	0.2000	0.8000
300	46	1.9730	0.2000	0.8000
400	46	1.9598	0.2000	0.8000
500	46	1.9519	0.2000	0.8000
600	46	1.9466	0.2000	0.8000
700	46	1.9428	0.2000	0.8000
800	46	1.9399	0.2000	0.8000
900	46	1.9377	0.2000	0.8000
1000	46	1.9360	0.2000	0.8000
2000	46	1.9280	0.2000	0.8000
3000	46	1.9253	0.2000	0.8000
4000	46	1.9240	0.2000	0.8000
5000	46	1.9232	0.2000	0.8000
6000	46	1.9227	0.2000	0.8000
7000	46	1.9223	0.2000	0.8000

Table 4 System performance measures for $\lambda = 5$, $\mu = 10$, $p = 0.5$, $\sigma = 100$ and various Values of k

K	OCUT	MNCO	P_0	P_1
1	23	0.5246	0.5000	0.5000
2	18	0.3995	0.5000	0.5000
3	17	0.3579	0.5000	0.5000
4	16	0.3370	0.5000	0.5000
5	16	0.3245	0.5000	0.5000
6	15	0.3162	0.5000	0.5000
7	15	0.3102	0.5000	0.5000
8	15	0.3058	0.5000	0.5000
9	15	0.3023	0.5000	0.5000
10	15	0.2995	0.5000	0.5000
11	15	0.2972	0.5000	0.5000
12	15	0.2954	0.5000	0.5000
13	15	0.2938	0.5000	0.5000
14	14	0.2924	0.5000	0.5000
15	14	0.2912	0.5000	0.5000
16	14	0.2901	0.5000	0.5000
17	14	0.2892	0.5000	0.5000
18	14	0.2884	0.5000	0.5000
19	14	0.2877	0.5000	0.5000
20	14	0.2870	0.5000	0.5000
21	14	0.2864	0.5000	0.5000
22	14	0.2859	0.5000	0.5000
23	14	0.2854	0.5000	0.5000
24	14	0.2849	0.5000	0.5000
25	14	0.2845	0.5000	0.5000

Table 5 System performance measures for $\lambda = 5$, $\mu = 10$, $\sigma = 100$, $k = 5$ and various Values of p

p	ocut	MNCO	p0	p1
1.000000	15	0.3000	0.5000	0.5000
0.500000	16	0.3245	0.5000	0.5000
0.250000	16	0.3371	0.5000	0.5000
0.125000	16	0.3435	0.5000	0.5000
0.062500	16	0.3468	0.5000	0.5000
0.031250	16	0.3484	0.5000	0.5000
0.015625	16	0.3492	0.5000	0.5000
0.007813	16	0.3496	0.5000	0.5000
0.003906	16	0.3498	0.5000	0.5000
0.001953	16	0.3499	0.5000	0.5000
0.000977	16	0.3499	0.5000	0.5000
0.000488	16	0.3500	0.5000	0.5000
0.000244	16	0.3500	0.5000	0.5000
0.000122	16	0.3500	0.5000	0.5000
0.000061	16	0.3500	0.5000	0.5000
0.000031	16	0.3500	0.5000	0.5000
0.000015	16	0.3500	0.5000	0.5000
0.000008	16	0.3500	0.5000	0.5000
0.000004	16	0.3500	0.5000	0.5000
0.000002	16	0.3500	0.5000	0.5000
0.000001	16	0.3500	0.5000	0.5000
0.000000	16	0.3500	0.5000	0.5000
0.000000	16	0.3500	0.5000	0.5000
0.000000	16	0.3500	0.5000	0.5000
0.000000	16	0.3500	0.5000	0.5000
0.000000	16	0.3500	0.5000	0.5000
0.000000	16	0.3500	0.5000	0.5000

8. CONCLUSION

The Numerical study shows the changes in the system due to impact of retrial rate. The mean number of customers in the orbit decreases as retrial rate increases and it increases as arrival rate increases. The various special cases have been discussed and which are particular cases of this research work. This research work will be extended further by introducing various vacation policies, negative arrival and unreliable server etc.,

REFERENCE

[1] Artalejo. J.R (1999 a). A classified bibliography of research on retrial queues Progress in 1990-1999 *Top 7*, pp 187-211.

[2] Artalejo. J.R (1999 b). Accessible bibliography on retrial queues, *Mathematical and Computer Modelling 30*, pp 223-233.

[3] Artalejo J.R and A. Gomez-Corral (2008). Retrial Queueing systems-a computational Approach, *Springer.*

[4] Artalejo. J.R (2010). Accessible bibliography on retrial queues Progress in 2000-2009, *Mathematical and Computer Modelling Vol 51*, pp 1071-1081.

[5] Ayyappan, Gopal Sekar and Muthu Ganapathi Subramanian (2010), M/M/1 Retrial Queueing System with Erlang Type Service by Matrix Geometric Method, *International Journal of Computer, Mathematical Sciences and Applications,* Vol. 4, No. 3-4, pp 357-368.

[6] Ayyappan, Gopal Sekar and Muthu Ganapathi Subramanian (2010), M/M/1 Retrial Queueing System with Negative Arrival under Erlang-k Services by Matrix Geometric Method, *Applied Mathematical Sciences,* Vol. 4 No. 48, pp 2355-2367.

[7] Ayyappan, Gopal Sekar and Muthu Ganapathi Subramanian (2010), M/M/1 Retrial Queueing System with Vacation Interruptions under Erlang-k Service, *International Journal of Computer Applications,* Vol. 2, No. 2, pp 52-57.

[8] Ayyappan, Gopal Sekar and Muthu Ganapathi Subramanian (2010), M/M/1 Retrial Queueing System with Working Vacation under Erlang-k Service by Matrix Geometric Method, *International Journal of Statistics and Systems,* Vol. 5, No. 2, pp 229-242.

[9] Ayyappan, Gopal Sekar and Muthu Ganapathi Subramanian (2010), Single Server Retrial Queueing System with Second Optional Service under Erlang Services, *International Journal of Mathematical Archive,* Vol. 2, No. 1, pp 174-182.

[10] Ayyappan, Gopal Sekar and Muthu Ganapathi Subramanian (2010), M/M/1 Retrial Queueing System with Variable Service Rates under Erlang-k-Type Services, *Advances and Applications in Statistics,* Vol. 18, No. 2, pp 189-208.

[11] Ayyappan, Gopal Sekar and Muthu Ganapathi Subramanian (2010), M/M/1 Retrial Queueing System with Two Types of Vacation Policies under Erlang-k Service, *International Journal of Computer Applications,* Vol. 2, No. 8, pp 9-18.

[12] Falin G.I (1990). A survey of retrial queues. Queueing Systems 7,No.2,127-167

[13] Falin G.I and J.G.C. Templeton (1997),Retrial queues, Champman and Hall, London

[14] Srinivas R. Chakravarthy, Krishnamoorthy.A and Joshua.J.C (2006), Analysis of a multi server retrial queue with search of customers from the orbit,, Performance evaluation, Vol 63, No.8, pp 776-798

[15] Yang T and Templeton J.G.C (1987). A Survey on retrial queues. Queueing systems, 2, No. 201-233.