

Performance Inference Engine (PIE) -- Deducing *More* Performance using *Less* Data

Susmit H. Patel, *Member, IEEE*

Abstract--The Performance Inference Engine (PIE) provides a basis for mathematical analytic technique to deduce performance measures from the transactional data for the given system. The transactional data are the customer interarrival and service time epochs in a given busy period. Such data are usually made available in some computer recorded digital format. From the given transactional data, queuing sample paths can be derived and both transient (time-dependent) and steady-state (time-averaged) performance measures could be derived using the PIE technique. The performance measures can be deduced in cases where only partially observable transactional data are available. The PIE method is being applied to derive new performance measures in telecommunications networks using transactional data. These results have been developed to include current and emerging network traffic models such as for circuit, packet and cell networks. This work is directly applicable to deriving end-to-end network performance measures by analyzing the network into a number of small interconnected systems, each operating with its own set of transactional data. Other performance-related measures such as Quality of Service (QoS), Service Level Agreement (SLA) could also be derived using the PIE technique.

Index Terms--Performance, queuing networks, inference, QoS, SLA, telecommunications traffic models.

I. INTRODUCTION

There is an abundant amount of data being generated by various modern communications and computer systems. These systems are (usually) well equipped by some type of internal digital recording device. The recording device could record an extensive amount of data. Such data may consist of, for example, initiation/termination time of each transaction, length or duration of transaction, type of transaction, cost of transaction, etc. For example, in case of an Automated Teller Machine at a bank; the bank manager, at the end of the day, can tell from the transactional log how many customers came to the bank, at what time and what type of transactions were made, etc. In order to understand the behavior (congestion, waiting time, etc.) of the system (bank teller machine), the collected data must be further refined and analyzed. Another example is that of pressure sensitive cables at traffic lights. Such system may record the time of arrival and departure of cars passing through the traffic lights. Data collected by the system may aid the analyst to understand traffic congestion and backup.

In case of the telecommunications networks; a router, a switch or a PBX may record the packet or call arrival and/or departure times, holding times (or packet lengths), number of calls blocked, packet delays/discards, etc.

When the raw form of this data are made available to the analyst, one can derive extensive performance measures using the techniques described in this paper. In many situations, the data made available to the analyst are partial or incomplete. The Performance Inference Engine (PIE), illustrated in this paper, works on the partial data to derive performance measures of interest. In addition to basic performance numbers, extensive performance including transient (time-dependent) and steady-state (time-averaged) measures can also be derived using the PIE technique.

II. SIMPLE QUEUING MODEL ILLUSTRATED

In order to understand the PIE technique, one has to first understand the basic queuing model. Consider a simple queuing model as shown in Figure 1 below. In this model, we have a single server system. The capacity of this system (waiting room) is assumed to be infinite. Customers from outside arrive to the system and form a single queue in front of the server. The queue discipline for the server is FCFS (First Come First Served). There are no priorities or preemption allowed in the system. The customer arrival rate and service rate for the system is assumed to be λ /unit time (per hour or per minute) and μ /unit time, respectively.

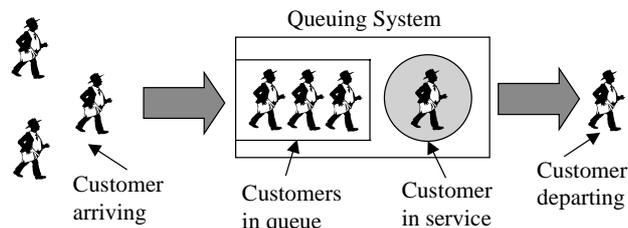


Figure 1. Simple Queuing Model

For illustrative purpose only, we consider sample data in which customers numbered 1 through 8 arrive to the system at the time epochs as shown in Table 1. Note that we have Arrival Time, Service Start Time and Service Stop Time shown in the shaded columns. If the quantities in these columns are known, then other performance measures can be determined.

Susmit H. Patel is a Principal Engineer at the MITRE Corporation in McLean, Virginia, USA. His e-mail address is spatel@mitre.org.

Let us see how we can proceed to derive the performance measures. We start with Arrival Time and Service Start Time for Customer 1 and note that since the Arrival Time = Service Start Time (10:05 in our example) for this customer, the customer must have found the server idle upon arrival and hence there is no queue. Thus customer 1 begins the Busy Period. Customer 2 arrives at 10:07 but because the previous customer does not leave until 10:15, he/she waits till 10:15 before invoking service at that time. The number in queue after the arrival of the 2nd customer is 1. Similarly, the 3rd customer arrives at 10:10 and immediately waits in the queue thus increasing the queue length to 2 until 10:15 when the queue length reduces back to 1. One can easily derive the Time in Queue which equals to Service Start Time – Arrival Time, and Time in System which equals Time in Queue + Service Time (Service Stop Time – Service Start Time). The busy period ends at 10:20 when the system becomes completely empty. There are no more customers in the queue or in service at that time.

The process repeats itself starting at 10:25. In the 2nd busy period, now we have different set of customers arriving, waiting and being served. The process alternates between the busy period and the idle period. The combination of alternating busy and idle periods is called a cycle. Each cycle starts and ends with busy period (equivalent to saying that upon arrival the customer finds the server idle). The idle period begins at the departure of the last customer in the busy period.

Note that, until now, we have not said anything about the distribution of the interarrival or service times. One can derive averages of certain performance measures by taking a number of these cycles and taking care of appropriate proportionality constants.

Note that we only considered few cycles consisting of a busy period and an idle period to examine. In real-world example, there will be a number of busy cycles (see Figure 2) consisting of alternating busy and idle periods in a typical queuing system. One would take the results of individual busy cycle, normalize it with respect to number of customers and time duration to come up with overall system-wide measure during the time period identified for the analysis. When sufficiently large (infinite) number of cycles are analyzed, one would get performance measures that would be close to the steady-state results. The steady-state results for infinite time duration of many cycles will reach limits so that taking furthermore samples will not change the results. These results are referred to as the analytic results.

Now we generalize the problem. We want to analyze one busy cycle consisting of a busy period and an idle period as illustrated by the sample path in Figure 3. We consider a series of customers carrying successive numbers, 1, 2, 3, ..., etc., arrive at instants $a_1 < a_2 < a_3 \dots$ at the FCFS queue and begin the service at times $y_1 < y_2 < y_3 \dots$ immediately after finding the server free. Upon completing the service, they depart from the system at instants $d_1 < d_2 < d_3 \dots$. The

server deals with them in same order; the customers line up in the queue and it is always the one at the head of line who is being served. All customers are treated with equal priority. This is a single server system with infinite capacity and no jockeying or balking is allowed in the system.

We are considering the points $t = a^-$ and $t = b^-$ when immediately upon arriving the customers find the system empty. Note that we let $N(t)$ be the number of customers in the system (in queue plus in service) at time t and consider the time interval $[a, b]$ such that $N(a) = N(b) = 0$.

For this time interval let $T(n)$ be the time spent by the system in the state $N(t) = n$. Let $\alpha(n)$ be the number of arrivals (indicated by α) of customers when the system size is n and $\beta(n)$ is the number of departures (indicated by β) when the system state is $n + 1$, still in the same interval. The marked regions in Figure 3 are shown for $n = 3$ and $n = 4$.

Since $N(a) = N(b) = 0$, it is necessary that:

$$\alpha(n) = \beta(n + 1)$$

Let $T = b - a$ be the duration of this interval. Hence:

$$\frac{\alpha(n)}{T} = \frac{\beta(n + 1)}{T}, n = 0, 1, 2, \dots$$

Table 2 shows the detailed account of arrivals, departures and time duration for each state based on the time epochs for this arbitrary interval.

Note that for system size $n = 0$, there is no departure. The sum of all the time duration give us the original time interval $[b - a]$ we are analyzing. In this sample interval, the points a and a_1 are same. Let $p(n)$ be the proportion of time spent in state n , in the interval in question:

$$p(n) = \frac{T(n)}{T} \text{ and } p(n + 1) = \frac{T(n + 1)}{T}.$$

Hence substituting for T , we get:

$$p(n) \frac{\alpha(n)}{T(n)} = p(n + 1) \frac{\beta(n + 1)}{T(n + 1)}.$$

To simplify the above formulas, let $\lambda(n)$ be the number of arrivals in unit time when the system size is n in the interval $[a, b]$:

$$\lambda(n) = \frac{\alpha(n)}{T(n)}$$

and let $\mu(n)$ be the number of departures in unit time:

$$\mu(n) = \frac{\beta(n)}{T(n)}.$$

Hence we have recurrence:

$$p(n + 1) = \left[\frac{\lambda(n)}{\mu(n + 1)} \right] p(n), n = 0, 1, 2, \dots$$

of which the solution is:

$$p(n) = p(0) \prod_{i=1}^n \frac{\lambda(i-1)}{\mu(i)}, n = 1, 2, \dots \quad (1)$$

Table 1. Illustration of Transactional Data for a Simple Queuing Model

Customer Number	Arrival Time	Arrival Finds Server Idle/Busy?	Number in Queue after Arrival	Service Start Time	Service Stop Time	Time in Queue	Time in Service	Time in System	Comment
1	10:05	Idle	0	10:05	10:15	0:00	0:10	0:10	10:05 Begin busy period1
2	10:07	Busy	1	10:15	10:18	0:08	0:03	0:11	
3	10:10	Busy	2	10:18	10:20	0:08	0:02	0:10	10:20 End busy period1
4	10:25	Idle	0	10:25	10:30	0:00	0:05	0:05	10:25 Begin busy period2
5	10:28	Busy	1	10:30	10:42	0:02	0:12	0:14	
6	10:29	Busy	2	10:42	10:45	0:13	0:03	0:16	
7	10:44	Busy	1	10:45	10:50	0:01	0:05	0:06	
8	10:46	Busy	1	10:50	10:55	0:04	0:05	0:09	10:55 End busy period2

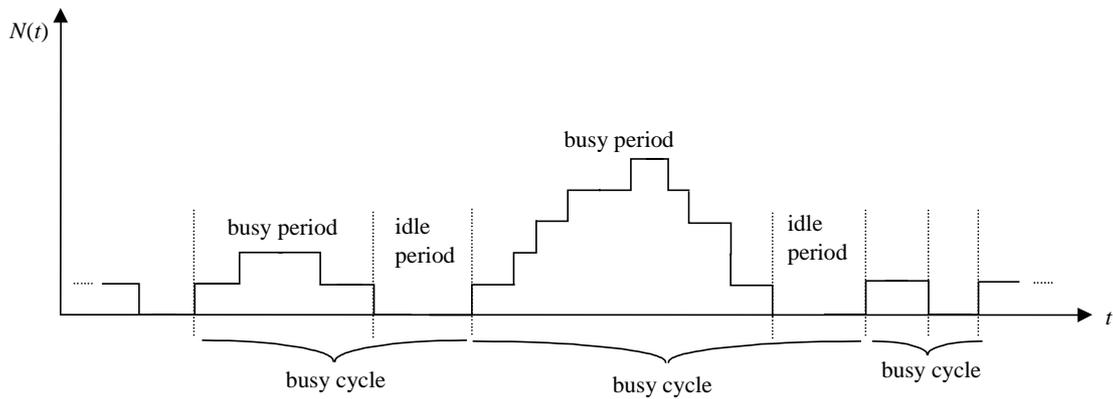


Figure 2. Illustration of Cycles with Alternating Busy and Idle Periods

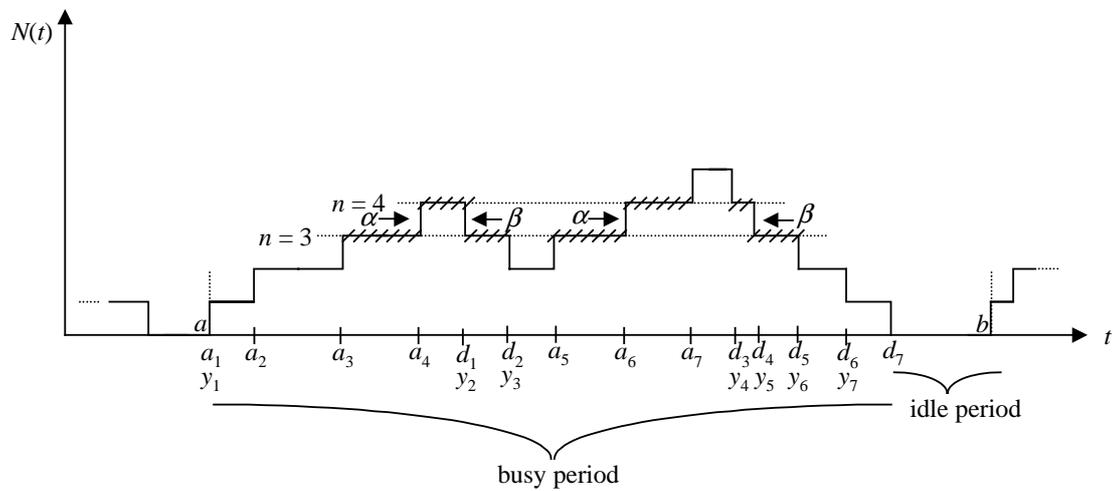


Figure 3. Illustration of a Sample Path

Table 2. Quantities $\lambda(n)$ and $\mu(n)$ for the Sample Path

System size (n)	Number of arrivals = $\alpha(n)$ when the system size is n	System size ($n + 1$)	Number of departures = $\beta(n + 1)$ when the system size is $n + 1$	Time duration $T[n]$ in the interval $[b - a]$ when the system size is n
0	1	1	1	$[b - d_7]$
1	1	2	1	$[a_2 - a_1] + [d_7 - d_6]$
2	2	3	2	$[a_3 - a_2] + [a_5 - d_2] + [d_6 - d_5]$
3	2	4	2	$[a_4 - a_3] + [d_2 - d_1] + [a_6 - a_5] + [d_5 - d_4]$
4	1	5	1	$[d_1 - a_4] + [a_7 - a_6] + [d_4 - d_3]$
5	0	6	0	$[d_3 - a_7]$
≥ 6	0	≥ 7	0	0

Therefore the proportions of time spent in each state must satisfy (1) and this formula will be applicable to all time intervals, which start and finish with an 'empty' state of the system. The quantity $p(0)$ can easily be determined since we must obtain:

$$p(n) = p(0) \prod_{i=1}^n \frac{\lambda(i-1)}{\mu(i)}$$

which gives:

$$p(0) = \frac{\sum_{n=0}^{\infty} \prod_{i=1}^n \frac{\lambda(i-1)}{\mu(i)}}{\sum_{n=0}^{\infty} \prod_{i=1}^n \frac{\lambda(i-1)}{\mu(i)}} \quad (2)$$

It must be emphasized that the quantities $\lambda(n)$ and $\mu(n)$ are measurable experimentally; and can be calculated (refer to Table 2) based on time epochs available from the interval. But formulas (1) and (2) do not predict the system's behavior outside the precise interval which starts at time a and finishes at time b . Nevertheless, such property occurs in certain probabilistic models and which can be used to predict behaviors in known statistical conditions [4].

When it is desirable to predict the performance of a system outside an observation period during which all the data concerned are accessible, it becomes necessary to make certain assumptions concerning its behavior. The probabilistic assumptions, which we shall make in here and in those which, follow lead to predictions concerning the systems, which we shall analyze. The link between these probabilistic assumptions and measurements on actual systems is established with aid of statistics.

Now we will assume that the FCFS queue being studied here is formed in which the times between successive arrivals and service times are mutually independent random variables distributed according to the exponential distribution.

The results of previous discussion concern a particular interval $[a, b]$ and a deterministic behavior. The results, which we obtain in here and others, that follow, concern an infinite period of time and (probable) set of realizations.

Because of the memoryless property of the exponential distributions, it allows the convergence of the problem when infinite busy cycles are taken into account.

Now we assume that the duration (or time intervals) $I_1 = a_1 - 0, I_2 = a_2 - a_1, \dots$ are random variables, that is they consist of quantities whose exact values are not known but for which a probability distribution can be defined. The variables S_1, S_2, S_3, \dots represent successive service times and are also assumed to be of random nature.

We assume that the time intervals between successive arrivals, or interarrival times, I_1, I_2, I_3, \dots are all distributed according to the exponential distribution:

$$\Pr\{I_j < x\} = 1 - e^{-\lambda x}, \quad j \geq 1,$$

the same assumption is made for the service times:

$$\Pr\{S_i < x\} = 1 - e^{-\mu x}, \quad i \geq 1,$$

where λ and μ (the parameters of the distributions) are real positive and finite. On the other hand we suppose that for $i \neq j$,

$$\Pr\{I_i < x \text{ and } I_j < y\} = \Pr\{I_i < x\} \Pr\{I_j < y\}$$

that is to say the interarrivals are independent. We also assume that the service times are independent of each other and that they are independent of the interarrivals.

The exponential distribution has one particularly interesting property or Memorylessness, which is one of the factors explaining its popularity. This simply means that suppose we wish to know the distribution of $X - y$ knowing that $X > y$ since the event has not occurred at time $t = y$. We calculate:

$$\begin{aligned} \Pr\{X - y < x \mid X > y\} &= \frac{\Pr\{y < X < y + x\}}{\Pr\{X > y\}} \\ &= \frac{1 - e^{-\lambda(y+x)} - (1 - e^{-\lambda y})}{e^{-\lambda y}} \\ &= 1 - e^{-\lambda x} = \Pr\{X < x\} \end{aligned}$$

and we discover that the event has not occurred up to time t allows us to establish simply that $X - y$ has the same distribution as X . This is called the Markovian or 'Memoryless' property of the exponential distribution.

In fact it can be proved that if a continuous positive random variable has the above property, then its distribution is exponential.

The queue being studied has been defined as a system in which the times between successive arrivals and service times are mutually independent random variables distributed according to the exponential distribution. We approach its analysis in a manner analogous to that adopted previously and we let $N(t)$ be the instantaneous number of customers in the system (including in the queue and in service).

Consider the existence of distinct times a and b at which the system is empty ($N(a) = N(b) = 0$), but chosen such that $N(t) > 0$ for $t = a^+$ and $t = b^+$: that is at times a^+ and b^+ , an arrival occurs. We require also that the system becomes empty once between a and b .

Let $\pi_{i,j}$ be the probability of passing from state i to state j after an arrival at or departure from the system. Let $m(n)$ be the mean number of times that the system passes through state n between a and b . Since to reach state n , one must arrive either from state $n + 1$ or from state $n - 1$, one can immediately write for $n > 0$:

$$m(n) = \pi_{n+1,n}m(n+1) + \pi_{n-1,n}m(n-1)$$

a formula which will be verified if the passage from state i to state j depends only on i and j . This can be proved due to the 'memoryless' property of the exponential law of interarrivals and service durations.

Suppose that one is in a state $i > 0$ at a time t ; the passage to state $i + 1$ will occur at a time $t + y$ and between t and $t + y$ there will be no departures. The passage from i to $i - 1$ will be made in the same way. So due to the memoryless property:

$$\begin{aligned} \pi_{i,i+1} &= \sum_{y=0}^{\infty} P\{y < I < y + dy\} P\{S < y\}, \\ &= \sum_0^{\infty} dy \lambda e^{-\lambda y} e^{-\mu y} = \frac{\lambda}{\lambda + \mu}, \end{aligned}$$

where I and S denote the classes of interarrivals and service times, respectively. Also:

$$\begin{aligned} \pi_{i,i-1} &= \sum_{y=0}^{\infty} P\{I > y\} P\{y < S < y + dy\}, \\ &= \frac{\mu}{\lambda + \mu} \end{aligned}$$

One obtains for $n > 0$,

$$m(n) = \frac{\mu}{\lambda + \mu} m(n+1) + \frac{\lambda}{\lambda + \mu} m(n-1).$$

Let us examine $T(n)$ which will be the mean time spent in state n between a and b . $T(n)$ is the mean time spent in state n on each stay in that state multiplied by $m(n)$. But due to

the memoryless property, the time spent in a state $n > 0$ is distributed as the variable $\min(S, I)$ and its mean value is:

$$\sum_0^{\infty} y P\{y < \min(S, I) < y + dy\} = \{\lambda + \mu\}^{-1}$$

So one obtains the occurrence:

$$(\lambda + \mu)T(n) = \mu T(n+1) + \lambda T(n-1), \quad n > 0$$

of which the solution is:

$$T(n) = \frac{\sum_{k=0}^n \lambda \sum_{l=0}^{n-k} 1}{\sum_{k=0}^n \mu \sum_{l=0}^k 1} T(0).$$

But $m(0) = 1$ from the choice which we have made for the interval $[a, b]$, and $T(0)$ will be quite simply the mean time spent in state 0 in that interval. We shall exploit for one last time the memoryless property, which implies that:

$$T(0) = E[1] = \frac{1}{\lambda}.$$

Finally, we have:

$$T(n) = \frac{1}{\lambda} \frac{\sum_{k=0}^n \lambda \sum_{l=0}^{n-k} 1}{\sum_{k=0}^n \mu \sum_{l=0}^k 1}$$

and T , the mean duration of the interval in question, will be:

$$\sum_{n=0}^{\infty} T = E[b - a] = \frac{1}{\lambda} \sum_{n=0}^{\infty} \frac{\sum_{k=0}^n \lambda \sum_{l=0}^{n-k} 1}{\sum_{k=0}^n \mu \sum_{l=0}^k 1} = \frac{\lambda^{-1}}{1 - \frac{\lambda}{\mu}} \quad \text{if } \frac{\lambda}{\mu} < 1.$$

$$\sum_{n=0}^{\infty} T = \infty \quad \text{if } \frac{\lambda}{\mu} \geq 1.$$

The probability of state n is now defined by:

$$p(n) = \frac{T(n)}{T}$$

which is valid only if $T < \infty$, that is if $\lambda/\mu < 1$. So we have

$$p(n) = \frac{\sum_{k=0}^n \lambda \sum_{l=0}^{n-k} 1}{\sum_{k=0}^{\infty} \lambda \sum_{l=0}^{n-k} 1} - \frac{\lambda}{\mu} \frac{\sum_{k=0}^n 1}{\sum_{k=0}^{\infty} 1}$$

In fact the interval $[a, b]$ which we have chosen in this paragraph allows us to characterize completely the process $N(t)$ for all values of t and not only for $a < t < b$. To convince this, let us examine all the events which occur for $t > b$. The process $N(t)$ for $t > b$ will be determined by the duration of the first service, and by time t' of the first arrival which occurs after b .

Thus $(t' - b)$ is a time between two successive arrivals and so does not depend on $N(\tau)$ for $\tau < b$. Similarly, at time b a new service starts and its duration no longer depends on previous events. So it can be deduced that $N(t)$ for $t > b$ is independent of $N(\tau)$ for $\tau < b$: one can say that $t = b$ is a *regeneration point* of the process $N(t)$. If there exists an infinite series of successive instants b_1, b_2, b_3, \dots at which the system passes from the empty state to the state $n = 1$, it is evident that $N(t)$ takes over an interval $b_i < t < b_{i+1}$ will be independent of $N(t)$ over other intervals. But since successive intervals and services are identically distributed, one can say that for $i \neq j, \tau \geq 0$,

$$\Pr\{N(\tau + b_i), b_i + \tau < b_{i+1}\} = \Pr\{N(\tau + b_j), b_j + \tau < b_{j+1}\}.$$

that is the process $N(t)$ repeats itself in successive intervals $[b_i, b_{i+1}]$. A process, which has this property, is called a *regenerative process*.

The practical conclusion is that the results which we have obtained for the intervals $[a, b]$ are in fact the same for all intervals $[b, b_{i+1}]$. This infinite series of instants $b_i, i > 1$, will exist if $E[b - a] < \infty$, that is $\lambda/\mu < 1$. It can also be proved, but more advanced mathematics would be required, that almost surely,

$$p(n) = \frac{T(n)}{E[b - a]} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \sum_0^{\tau} 1(N_t = n)$$

$$\text{where } 1(N_t = n) = \begin{cases} 1 & \text{if } N_t = n, \\ 0 & \text{otherwise.} \end{cases}$$

This formula indicates that $p(n)$ can be interpreted also as being, asymptotically, the proportion of time spent in state n (almost certainly).

So given the time interval $[a, b]$ and N customers who are served in this interval, we have the following time epochs information available from the transactional data.

a_i = arrival time for the i -th customer, $i = 1$ to N

y_i = service starting time for the i -th customer, $i = 1$ to N

d_i = departure time for the i -th customer, $i = 1$ to N

Now we can derive the following performance measures for this single interval:

1. Time in queue for the i -th customer = $y_i - a_i$
2. Time in service for the i -th customer = $d_i - y_i$
3. Time in system for the i -th customer = $d_i - a_i$
4. Customer arrival rate = $\lambda = N / (b - a)$
5. Customer service rate = $\mu = N / \sum_{i=1}^N (d_i - y_i)$
6. Traffic intensity = $\rho = \lambda/\mu$
7. Utilization (time of the interval when server was busy) = $\sum_{i=1}^N (d_i - y_i) / (b - a)$
8. Average (per Customer) Time in queue = $Wq = \sum_{i=1}^N (y_i - a_i) / N$
10. Average (per Interval) Number in queue = $Lq = \lambda * Wq = \sum_{i=1}^N (y_i - a_i) / (b - a)$ (per Little's Law)
11. Average (per Customer) Time in system = $W = \sum_{i=1}^N (d_i - a_i) / N$
12. Average (per Interval) Number in system = $L = \lambda * W =$

$$\sum_{i=1}^N (d_i - a_i) / (b - a) \text{ (per Little's Law)}$$

13. Average (per Customer) Time in service = s.t.

$$= \sum_{i=1}^N (d_i - y_i) / N. \text{ Note that s.t.} = 1/\mu.$$

14. Probability that a server is busy = $p_b =$ Time in

$$\text{Service/Total Interval Time} = \sum_{i=1}^N (d_i - y_i) / (b - a)$$

15. Probability that a server is idle = $p_0 = 1 - p_b$

16. Throughput = service rate when server is working = $\mu * (1 - p_0)$

Note that all the performance measures we derive are for one cycle only. In order to expand these results to many cycles one must take appropriate scaling of customers and time interval to derive system-wide performance measures.

III. ROOTS OF PIE IN QUEUE INFERENCE ENGINE (QIE)

Now we are ready to introduce Performance Inference Engine (PIE) whose roots come from Queue Inference Engine or QIE [1]. The QIE is a set of algorithms for queue inference that allow the estimation of several key performance measures of interest for the queuing system.

One of the problems, addressed in the past, in such area has provided a mechanism by which queue lengths can be estimated based solely on the information about service commencements (initiations) and service completions (terminations) for a single busy period. The queue inference method is useful in situations where invisible queues exist or arrivals to the system are not observed directly but rather the service epochs are available for individual customers. Since the queue inference method gives transient performance results for the busy period, it is also useful in situations where one needs to take some corrective action for the system in real time given the current situation of the system under analysis. There are no assumptions made about service-time distributions or the number of servers in the system. Hence, queue inference analysis can be applied to model a number of problems where the service-time distributions are of general (G) type and/or the number of servers is unknown.

In the past, queue measure calculations using either the order statistics argument or the multi-dimensional integration methods have been reported in the literature ([1], [2]). But these derivation methods are very computational-intensive. Using the taboo probabilities for the discrete Markov chains simplifies the task of calculating the queuing measure [3].

For the case of single-server queue with N -customer departures during a busy period, Daley and Servi [3] showed that one can obtain the queue length distribution using taboo probabilities of the embedded Markov chain technique. This technique yields faster algorithm and more

accurate performance measures compared to the integration method and ordered-statistics methods described previously.

We expand upon the embedded Markov chain technique to obtain new and different performance measures for the basic model we will consider. Then we apply the results to a different set of problems.

Before we begin the derivation of new results for various queuing problems of interest using the PIE, we provide key notations and techniques leading to such derivation.

DERIVATION APPROACH AND COMPUTATIONAL DETAILS GIVEN DEPARTURE TIMES

We consider a time interval $[a, b]$ and let $N(t)$ be the number of customers in the system (in queue plus in service) at time t . We have two points $t = a^-$ and $t = b^-$ when immediately upon arriving to the system, the customers find the system empty. Note that $N(a) = N(b) = 0$.

We further assume that there is a positive queue length right after the start of the busy period and set $N(a^+) = 1$.

Before we begin the derivation of new results for various queue inference problems using embedded Markov chain technique, we provide the key notations and results of this method. This is a single-server FCFS queue in which N customers depart (or get served) during the interval time $[a, b]$. We assume that the arrival process is Poisson and the busy period starts at time epoch a when an arriving customer finds an empty server. For this N -customer busy period, we know the following time epochs:

- a – epoch of the time interval start/busy period
- $\{d_1, d_2, d_3, \dots, d_N\}$ - departure time epochs for customers 1 through N of the busy period
- $\{y_1, y_2, y_3, \dots, y_N\}$ - service starting time epochs for customers 1 through N of the busy period
- $\{d_1, d_2, d_3, \dots, d_N\}$ - departure time epochs for customers 1 through N of the busy period (note d_N epoch also denotes the end of busy period)
- b – epoch of the time interval end

The $\{a_1, a_2, a_3, \dots, a_N\}$ - arrival time epochs for customers 1 through N of the busy period are unknown. Thus, a_1 is the arrival time for the first customer who finds an idle server and immediately starts the service. Hence, $a_1 = y_1$.

We further know that the remaining $N - 1$ customers were queued and their service starting times during the busy period are known to be $\{y_2, y_3, \dots, y_{N-1}, y_N\}$. Since the customer in queue immediately join for the service as the previously-served customer departs from the system, the service starting times for customers 2 through N correspond to the departure times of customers 1 through $N-1$. Accordingly,

$$\{y_2, y_3, \dots, y_{N-1}, y_N\} \equiv \{d_1, d_2, \dots, d_{N-2}, d_{N-1}\}$$

Or more generally,

$$y_2 = d_1, y_3 = d_2, \dots, y_{N-1} = d_{N-2}, y_N = d_{N-1}.$$

When the arrival process is Poisson, the standard approach to studying the stochastic behavior of such a queue is to consider the distribution $\Pr\{N(d_r) = j\}$ at system departure epochs ($r = 1, 2, \dots, N - 1, N$). Such problems (i.e. with Poisson arrivals and general service discipline) are typically classified and analyzed as the $M/G/c$ type problems.

The study of a busy period is equivalent to studying $N(d_r)$ for $r < N$ subject to $N(a_1) = 1, N(d_N) = 0$, and to the event $N(d_s) = j$ ($j > 1$) for any $s = 1, 2, \dots, N - 1$ being a taboo event. We use the following notation to denote number in queue between the two epochs during the busy period. Such notation also denotes any subset of the complement of a general busy period taboo event.

$$A^{r_1, r_2} = \{N(d_s) > 0 : s = r_1, \dots, r_2\}.$$

We are interested in the distribution of $N(d_r)$ under the assumption that d_r is an epoch of departure in a busy period in which exactly N customers are served ($r = 1, 2, \dots, N-1, N$). Because we assume that this is a single-server queue, we know that the service times for each customer is simply the difference between next customer's service starting time and his own service starting time. So we denote the service times as S_1, S_2, \dots, S_N . And thus,

$$\begin{aligned} \text{for } r = 1 \text{ we have } S_1 &= d_1 - a_1 = d_1 \text{ and} \\ \text{for } r = 2, \dots, N \text{ we have } S_r &= d_r - d_{r-1} \\ \text{or } d_r &= S_1 + \dots + S_r = d_{r-1} + S_r. \end{aligned}$$

Since we know the $\{d_r\}$, we also know $\{S_r\}$. We want the conditional distribution for $N(d_r)$ at departure epochs given there are N customers who departed during the busy period. We also notice that at the epochs $\{d_1, d_2, \dots, d_{N-1}\}$, two events – a simultaneous service initiation and a departure from the queue – occur. We observe that $N(d_r) = N_r = N(d_r - 0)$ by the left-continuity of $N(\cdot)$, and $N(d_r + 0) = N(d_r) - 1$. Then we seek, for $j = 1, 2, \dots$ and $r = 1, 2, \dots, N - 1$.

$$\begin{aligned} Q_{j|N}^r &\equiv \Pr\{N_r = j \mid N_0 = 1, N_s \geq 1 (s = 1, \dots, N - 1), N_N = 0\} \\ &= \Pr\{N_r = j \mid N_0 = 1, A^{1, N-1}, N_N = 0\} \\ &= \frac{\Pr\{N_r = j, A^{1, N-1}, N_N = 0 \mid N_0 = 1\}}{\Pr\{A^{1, N-1}, N_N = 0 \mid N_0 = 1\}}. \end{aligned}$$

The notation $Q_{j|N}^r$ denotes the probability of having j customers in the queue at (or just before) the r -th epoch given that there are N customers who departed during the busy period. The numerator and denominator here are both examples of taboo probabilities for the discrete-time Markov chains on the non-negative integers $\{N_r\}$. Specifically here, we use the notation:

$$E[A(t)] = \frac{\sum_{r=1}^N d_r - t}{\sum_{r=1}^N d_r - d_{r-1}} \sum_{r=1}^N E[A(d_{r-1})] + \frac{\sum_{r=1}^N t - d_{r-1}}{\sum_{r=1}^N d_r - d_{r-1}} \sum_{r=1}^N E[A(d_r)]$$

for $r = 1, 2, \dots, N-1$.

2. Expected queue length that the r -th departure sees: at d_r - (or just before d_r) =

$$E[Q(d_r)] = \sum_{j=1}^{N-r} j Q_{j|N}^r \text{ for } r = 1, 2, \dots, N-1.$$

The transient queue length $E[Q(t)]$ at any general time t (within $(0, d_{N-1})$ is calculated as:

$$E[Q(t)] = \frac{\sum_{r=1}^N d_r - t}{\sum_{r=1}^N d_r - d_{r-1}} \sum_{r=1}^N E[Q(d_{r-1})] + \frac{\sum_{r=1}^N t - d_{r-1}}{\sum_{r=1}^N d_r - d_{r-1}} \sum_{r=1}^N E[Q(d_r)]$$

for $r = 1, 2, \dots, N-1$.

3. Customer arrival rate over one interval = $\lambda = N / (b - a)$

4. Time-averaged queue length for the busy period =

$$L_q = \frac{1}{b - a} \int_0^b E[Q(t)] dt.$$

5. Time-averaged wait in queue for the interval (using Little's Law) =

$$W_q = \frac{\sum_{i=1}^N b - a}{\sum_{i=1}^N N} \sum_{i=1}^N L_q.$$

6. Customer service rate = $\mu = N / \sum_{i=1}^N (d_i - y_i)$

7. Time in queue for the i -th customer = $y_i - \hat{a}_i$ where \hat{a}_i is an estimate of the arrival time derived as:

$$\hat{a}_i = \min\left\{a + \frac{1}{\lambda}(i), y_i\right\}$$

8. Time in service for the i -th customer = $d_i - y_i$

9. Time in system for the i -th customer = $d_i - \hat{a}_i$

10. Traffic intensity = $\rho = \lambda \mu$

11. Utilization (time of the interval when server was busy) =

$$\sum_{i=1}^N (d_i - y_i) / (b - a)$$

12. Average (per Customer) Time in system = $W =$

$$\sum_{i=1}^N (d_i - \hat{a}_i) / N$$

13. Average (per Interval) Number in system = $L = \lambda * W =$

$$\sum_{i=1}^N (d_i - \hat{a}_i) / (b - a) \text{ (per Little's Law)}$$

14. Average (per Customer) Time in service = s.t.

$$= \sum_{i=1}^N (d_i - y_i) / N. \text{ Note that s.t.} = 1/\mu.$$

15. Probability that a server is busy = $p_b =$ Time in

$$\text{Service/Total Interval Time} = \sum_{i=1}^N d_i - y_i / (b - a)$$

16. Probability that a server is idle = $p_0 = 1 - p_b$

17. Throughput = service rate when server is working = $\mu * (1 - p_0)$

Having provided the sufficient background and terminology for the PIE, we are ready to shown PIE application to queuing networks and telecommunications traffic problem.

IV. PIE FOR MODELS OF QUEUEING NETWORKS

Queuing networks can be described as a group of nodes (k of them) where each node represents a service facility of some kind. Each node may have one or more servers. One may denote, say c_i servers at node i , $i = 1, 2, \dots, k$. In the most general case, the customers may arrive from outside the system to any node and may depart from the system from any node. Thus the customers may enter the system at some node, traverse from node to node in the system, and depart from some node, not all customers necessarily entering and leaving at the same nodes, or taking the same path once having entered the system. Customers may return to nodes previously visited, skip some nodes entirely, and even choose to remain in the system forever.

We will mainly be concerned with queuing networks with the following characteristics:

- (i) Arrivals from the "outside" to node i follow Poisson process with mean rate γ_i .
- (ii) Service (holding) times for each channel at node i are independent and exponentially distributed with parameter μ_i .
- (iii) The probability that a customer who has completed service at node i will go to next node j (routing probability) is r_{ij} (independent of the state of the system).

A. 2-NODE TANDEM QUEUEING NETWORKS – WITH UNLIMITED CAPACITY AT EACH NODE – SAME N CUSTOMERS TRAVERSE THE NETWORK SEQUENTIALLY

We start by considering the simplest example for a two-node network. We consider a two-node tandem network having a single server at each node with unlimited capacity. We assume that the arrival process to the network is *Poisson*. Customers queue in front of the server at each node and are served on a FCFS basis. As shown in Figure 4, only external arrivals to the network at node 1 are allowed. Customers may not depart the network at node 1 and must enter queue in front of node 2, before being served and departing from node 2.

Note that no external customers are allowed in this system and we have the following data available for the interval time $[a, b]$.

- a – epoch of the time interval start/busy period
- $\{d_{i1}, d_{i2}, d_{i3}, \dots, d_{iN}\}$ - departure time epochs for customers 1 through N at node i of the busy period
- b – epoch of the time interval end

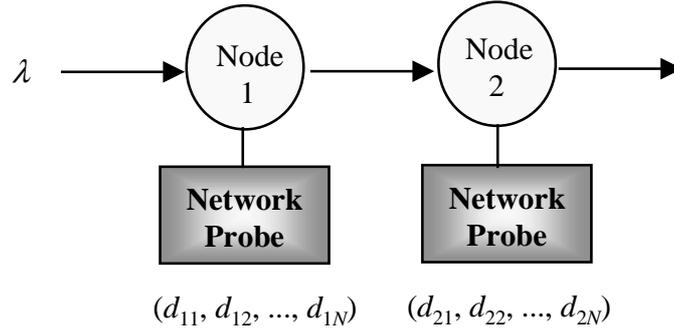


Figure 4. Two-node Tandem Queuing Network

Note that the busy period for this network starts at the arrival of the first customer. The busy period ends when the last customer leaves Node 2 at epoch d_{2N} . We further assume that there are (passive) Network Probes in the system that captures the departure time epochs for each customer (packet). Now we want to derive performance measures given this partial data.

Using the PIE technique, we can easily derive the following performance measures. The Transient Conditional Probability Transition Matrix (TCPTM) for Node 1 can be written as:

$$\{ Q_{(1)jN}^r \} =$$

	1	2	...	N-1	
→ j					
1	$\sum_{i=1}^N Q_{(1)1N}^1$	$Q_{(1)2N}^1$.	$Q_{(1)2N}^1$	Σ
2	$\sum_{i=1}^N Q_{(1)1N}^2$	$Q_{(1)2N}^2$.	$Q_{(1)N-2N}^2$	-
↓ r .	$\sum_{i=1}^N Q_{(1)1N}^3$.	.	-	-
.	$\sum_{i=1}^N Q_{(1)1N}^r$.	-	-	-
r	$\sum_{i=1}^N Q_{(1)1N}^r$	-	-	-	Σ

Note that all the values of above matrix can be determined using the departure epoch values $\{d_{11}, d_{12}, d_{13}, \dots, d_{1N}\}$ at Node 1.

For Node 2, we have more information than just the departure times for Node 2. Since we are dealing with a system where customers from Node 1 must go to Node 2 before leaving. All the departure epoch information for Node 1 becomes Arrival epoch information for Node 2. Thus

$$d_{11} = a_{21}, d_{12} = a_{22}, \dots, d_{1N} = a_{2N}.$$

1. Expected queue length that the r -th departure sees at Node 1 at d_{1r} - (or just before d_r) =

$$E[Q(d_{1r})] = \sum_{j=1}^{N-r} j Q_{(1)jN}^r \quad \text{for } r = 1, 2, \dots, N-1,$$

where the value of $Q_{(1)jN}^r$ can be obtained from above TCPTM.

The transient queue length at Node 2 can be determined accurately using all the data that is made available by the Network Probes. Accordingly,

1. Customer arrival rate for the Network over one interval = $\lambda = N / (b - a)$

2. Customer service rate at Node 1 = $\mu_1 = N / \sum_{i=1}^N (d_{1i} - y_{1i})$

Note that $\{y_{11}, y_{12}, y_{13}, \dots, y_{1N}\} = \{a, d_{11}, d_{12}, \dots, d_{1N-1}\}$.

3. Customer service rate at Node 2 = $\mu_2 = N / \sum_{i=1}^N (d_i - y_i)$

Note that the service rate μ for the network does not equal $\mu_1 + \mu_2$ because of overlapping times between the servers during which both servers were functioning.

4. Customer service rate for the Network over one interval = $\mu = N / (d_{2N} - a)$

5. Time-averaged queue length for the busy period =

$$L_q = \frac{1}{b-a} \sum_0^b E[Q(t)] dt.$$

6. Time-averaged wait in queue for the interval (using Little's Law) =

$$W_q = \frac{\sum b - a}{\sum N} \frac{\sum L_q}{\sum L_q}.$$

7. Time in queue for the i -th customer = $y_i - \hat{a}_i$ where \hat{a}_i is an estimate of the arrival time derived as:

$$\hat{a}_i = \min\left\{ \left(a + \frac{1}{\lambda}(i)\right), y_i \right\}$$

8. Time in service for the i -th customer at each Node = $d_i - y_i$

9. Time in system for the i -th customer at each Node = $d_i - \hat{a}_i$

10. Traffic intensity = $\rho = \lambda/\mu$

11. Utilization (time of the interval when each server was busy) =

$$\sum_{i=1}^N (d_i - y_i) / (b - a)$$

12. Average (per Customer) Time in system = $W =$

$$\sum_{i=1}^N (d_i - \hat{a}_i) / N$$

13. Average (per Interval) Number in system = $L = \lambda * W =$

$$\sum_{i=1}^N (d_i - \hat{a}_i) / (b - a) \text{ (per Little's Law)}$$

14. Average (per Customer) Time in service = s.t.

$$= \sum_{i=1}^N (d_i - y_i) / N. \text{ Note that s.t.} = 1/\mu.$$

15. Probability that a server is busy = $p_b =$ Time in

$$\text{Service/Total Interval Time} = \sum_{i=1}^N d_i - y_i / (b - a)$$

16. Probability that a server is idle = $p_0 = 1 - p_b$

17. Throughput = service rate when server is working = $\mu * (1 - p_0)$

B. *k*-NODE TANDEM QUEUEING NETWORKS – WITH UNLIMITED CAPACITY AT EACH NODE – SAME *N* CUSTOMERS TRAVERSE THE NETWORK SEQUENTIALLY

One can easily generalize the above results to *k*-node tandem network where we assume that the arrival process to the network is *Poisson*. Customers queue in front of the server at each node and are served on a FCFS basis. Only external arrivals to the network at node 1 are allowed. Customers may not depart the network at node 1 and must enter queue in front of node 2, 3, ..., *k* before being departing from node *k*.

Note that no external customers are allowed in this system and we have the following data available for the interval time [*a*, *b*].

- *a* – epoch of the time interval start/busy period
- $\{d_{i1}, d_{i2}, d_{i3}, \dots, d_{iN}\}$ – departure time epochs for customers 1 through *N* at node *i* of the busy period and *i* = 1, 2, 3, ..., *k*.
- *b* – epoch of the time interval end

Results similar to 2-Node examples can be obtained using the PIE technique.

C. 3-NODE TREE QUEUEING NETWORKS – WITH UNLIMITED CAPACITY AT EACH NODE – SAME *N* CUSTOMERS TRAVERSE THE NETWORK SEQUENTIALLY

We consider the example of 3-node Tree Networks as shown in Figure 5. We further assume that there is unlimited capacity at each node (no blocking) and the traffic splits up in a probabilistic manner at Node 1. This is an important assumption, because if we were to assume

deterministic splitting of traffic from one node to another, no external customers allowed except at the first node)

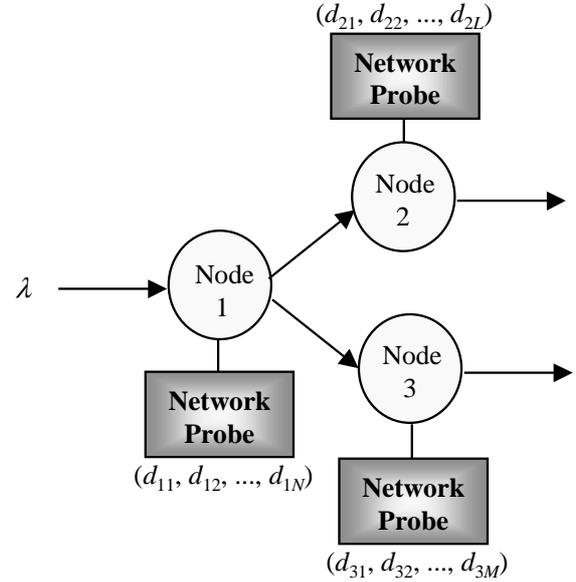


Figure 5. Example of a 3-Node Tree Network

We further assume that we have the following transactional data available for the time interval [*a*, *b*].

- *a* – epoch of the time interval start/busy period
- $\{d_{11}, d_{12}, \dots, d_{1N}\}$ – departure time epochs for customers 1 through *N* at Node 1
- $\{d_{21}, d_{22}, \dots, d_{2L}\}$ – departure time epochs for customers 1 through *L* at Node 2
- $\{d_{31}, d_{32}, \dots, d_{3M}\}$ – departure time epochs for customers 1 through *M* at Node 3
- *b* – epoch of the time interval end

It is obvious that number of customers departing at Node 1 (*N*) equals the number of customers departing at Node 2 (*L*) and Node 3 (*M*), i.e. $N = L + M$. The Network Probes collect the departure time data in a manner similar to previously described. Each individual customer (packet) departure times are recorded and since no external customers are allowed in this system, they must also depart from the network either at Node 2 or at Node 3. The routing probability that customer at Node 1 will go to Node 2 is *p* and that of customer at Node 1 will go to Node 3 is (1 – *p*).

Now applying the PIE technique, one can get performance measures as described in the previous section. In addition to basic measures of performance, one can also obtain other network performance results. For example,

$$p = \text{the routing probability from Node 1 to Node 2} = \frac{L}{N}.$$

$$\text{Similarly, Routing probability from Node 1 to Node 3} = (1 - p) = \frac{M}{N}.$$

V. PIE FOR CIRCUIT/IP/ATM NETWORKS

As one can see from these examples, the PIE technique can easily be applied to understand the network performance employing the Circuit, IP and ATM protocols. These networks employ some fundamentally different paradigms – calls versus packets (variable length) versus cell (fixed length) and connectionless versus connection-oriented.

There are different performance measures that are of interest in these networks employing different protocols.

A. Circuit-switched Network

For example, in case of a Circuit-switched network, one is interested in understanding the blocking probability. We assume that we have the following data available from the transactional log. Such data can be obtained from PBX or Integrated Access Device (IAD) at the ingress point of the network. We break down our analysis in a single time interval and carry out the performance measures using PIE technique.

- a – epoch of the time interval start/busy period
- $\{y_1, y_2, y_3, \dots, y_N\}$ - service starting time epochs for customers 1 through N of the busy period
- $\{d_1, d_2, d_3, \dots, d_N\}$ - departure time epochs for customers 1 through N of the busy period (note d_N epoch also denotes the end of busy period)
- b – epoch of the time interval end

Furthermore, we assume that there are N customers and k trunks (servers). No queuing of calls is allowed. Calls are blocked (or lost) upon arriving if all servers are busy. The busy period is defined as when there is one or more call occupying the trunks. We assume that in 1 busy period, $\min\{N, k\}$ calls are served. We consider 3 cases depending upon the size of N and/or k .

Case 1: when $N = k$, the $(N+1)^{\text{st}}$ customer will be lost.

1. Call arrival rate over one interval = $\lambda = N / (b - a)$
2. Queue length and wait in queue for the interval = 0
3. Call service rate = $\mu = N / \sum_{i=1}^N (d_i - y_i)$
4. Call service time for the i -th call = $d_i - y_i$
5. Call system time for the i -th call = $d_i - y_i$
6. Traffic intensity = $\rho = \lambda / \mu$
7. Utilization (time of the interval when trunks are busy) = $\sum_{i=1}^N (d_i - y_i) / (b - a)$
8. Probability of blocking = $p_b = \text{Time in Service/Total Interval Time}$
9. Probability that one of the trunks is idle = $p_0 = 1 - p_b$
10. Probability of overflow of calls = $1 - p_b$

Case 2: when $N > k$, the $(k+1)^{\text{st}}$ customer will be lost.

1. Call arrival rate over one interval = $\lambda = k / (b - a)$
2. Queue length and wait in queue for the interval = 0
3. Call service rate = $\mu = k / \sum_{i=1}^k (d_i - y_i)$
4. Call service time for the i -th call = $d_i - y_i$
5. Call system time for the i -th call = $d_i - y_i$
6. Traffic intensity = $\rho = \lambda / \mu$
7. Utilization (time of the interval when trunks are busy) = $\sum_{i=1}^k (d_i - y_i) / (b - a)$
8. Probability of blocking = $p_b = \text{Time in Service/Total Interval Time}$
9. Probability that one of the trunks is idle = $p_0 = 1 - p_b$
10. Probability of overflow of calls = $1 - p_b$

Case 3: when $N < k$, the $(N + 1)^{\text{st}}$ customer will be lost.

1. Call arrival rate over one interval = $\lambda = N / (b - a)$
2. Queue length and wait in queue for the interval = 0
3. Call service rate = $\mu = N / \sum_{i=1}^N (d_i - y_i)$
4. Call service time for the i -th call = $d_i - y_i$
5. Call system time for the i -th call = $d_i - y_i$
6. Traffic intensity = $\rho = \lambda / \mu$
7. Utilization (time of the interval when trunks are busy) = $\sum_{i=1}^N (d_i - y_i) / (b - a)$
8. Probability of blocking = 0
9. Probability that one of the trunks is idle = 1
10. Probability of overflow of calls = 0.

B. IP Network

The case of IP-protocol based network has been already considered. In IP networks, one is interested in understanding the queuing and delays performance measures. This can be looked at various levels including per packet, per flow or per session.

We identify four such levels:

1. Network level
2. Session level
3. Flow level, and
4. Packet level.

At the Network level, there are multiple sessions occurring all over the network. These sessions are originating from a single user or multiple users. Each Session could have one or more flows that span from Source to Destination nodes taking different or same routes. Furthermore, each Flow itself consists of number of Packets (actual bits/bytes).

We start at the top-most (Network) level and consider a case of L links, S switches and N customers (Sessions) traversing the network. We can obtain results for its performance using the PIE technique given certain transactional data.

For simplicity, we denote L links and S Switches as Nodes and Sessions as Customers approaching these Nodes for service. The Network-level performance measures for N -sessions can be obtained as:

1. Session arrival rate for the Network over one interval = λ
 $= N / (b - a)$

2. Session service rate for the Network over one interval =
 $\mu = N / (d_N - a)$

3. Time-averaged queue length for the busy period =

$$L_q = \frac{1}{b - a} \sum_0^b \mathbb{E}[Q(t)] dt .$$

4. Time-averaged wait in queue for the interval (using Little's Law) =

$$W_q = \frac{\sum b - a}{\sum N} \frac{\sum L_q}{\sum L_q}$$

5. Time in queue for the i -th Session = $y_i - \hat{a}_i$ where \hat{a}_i is an estimate of the arrival time derived as:

$$\hat{a}_i = \min\left\{a + \frac{1}{\lambda}(i), y_i\right\}$$

6. Time in service for the i -th Session at Node = $d_i - y_i$

7. Time in system for the i -th customer at Node = $d_i - \hat{a}_i$

8. Traffic intensity = $\rho = \lambda/\mu$

9. Utilization (time of the interval when each server was busy) =

$$\sum_{i=1}^N (d_i - y_i) / (b - a)$$

10. Average (per Customer) Time in system = $W =$

$$\sum_{i=1}^N (d_i - \hat{a}_i) / N$$

11. Average (per Interval) Number in system = $L = \lambda * W =$

$$\sum_{i=1}^N (d_i - \hat{a}_i) / (b - a) \text{ (per Little's Law)}$$

12. Average (per Customer) Time in service = s.t.

$$= \sum_{i=1}^N (d_i - y_i) / N . \text{ Note that s.t.} = 1/\mu.$$

13. Probability that a Node is busy = $p_b =$ Time in

$$\text{Service/Total Interval Time} = \sum_{i=1}^N d_i - y_i / (b - a)$$

14. Probability that a Node is idle = $p_0 = 1 - p_b$

15. Throughput = service rate when Node is working =
 $\mu * (1 - p_0)$

Similarly, we can apply same technique to lower levels to obtain level-specific performance measures.

C. ATM Network

Now will briefly describe the use of PIE for ATM networks. Cell or fixed size packet (53 bytes) is the basic unit of information transfer in an ATM network. The behavior of traffic in an ATM network can be considered in several levels. We identify four such levels:

1. Network (or subscription level)
2. Call level
3. Burst level, and
4. Cell level.

At the Network level, a number of calls are carried and terminated. At the call level, a single call or session lasts for the duration of the connection between the end users. A call in turn can be partitioned into a sequence of alternate burst (ON) and silence (OFF) periods. These periods affect the burst level performance of an ATM network. During the ON period (Ton), a stream of ATM cells is emitted at regular intervals. During the off periods (Toff), no cells are emitted.

An important attribute of each level is its time scale. This is governed by the mean interarrival time of entities in that level during an activity in the upper level. Usually, time scales of different levels are substantially different. For example, for voice calls, the call level may be in the order of minutes or seconds, while burst level is in the order of milliseconds.

So we can follow the similar PIE technique that we have shown previously and can derive the performance measures at each level. Additional performance measures such as Cell delay and Cell jitter and/or Cell Loss Ratio can also be derived.

VI. SUMMARY

The Performance Inference Engine or PIE is an analytical tool to deduce performance measures for the given busy period using transactional data. The technique of PIE provides an in-depth examination of performance measures given full or incomplete transactional data. We illustrate several use of PIE to deduce performance measures, including examples of queuing networks and networks employing various protocols such as Circuit, IP and ATM.

We describe the scenario where one can implement (passive) Network Probes on the network that can capture customer (circuit, packet or cell) departure/arrival time data without affecting network. Using this data, one can derive transient and time-averaged performance measures. Several other uses of inference technique have also been reported in the literature (for example, see [5], [6], and [7]). Other performance measures such as QoS and SLA measures about the system and underlying network can also be derived using the PIE technique.

Furthermore, the PIE technique can also be applied to other (ex. non-Poisson) telecommunications traffic models.

REFERENCES

- [1] Larson, R., May 1990, "The Queue Inference Engine: Deducing Queue Statistics from Transactional Data," *Management Science*, Vol.36, No.5, pp. 586-601.
- Larson, R., August 1991, "The Queue Inference Engine: Addendum," *Management Science*, Vol.37, No.8, p. 1062.
- [2] Bertsimas, D. and Servi, L., 1992, "Deducing Queuing from Transactional Data: The Queue Inference Engine, Revisited," *Operations Research*, Vol.40, pp. S217-S228.
- [3] Daley, D. and Servi, L., 1992, "Exploiting Markov Chains to Infer Queue Length From Transactional Data," *Journal of Applied Probability*, Vol.29, pp.713-732.
- [4] Ross, S., 1983, *Stochastic Processes*, New York, NY: John Wiley & Sons.
- [5] Gawlick, R., 1990, "Estimating Disperse Network Queues: The Queue Inference Engine," *Computer Communications Review*, Association of Computing Machinery (ACM), No.20, pp.111-118.
- [6] Hall, S., 1992, "New Directions in Queue Inference for Management Implementation," Ph. D. Dissertation, Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology, Cambridge, MA.
- [7] Jones, L. and Larson, R., 1995, "Efficient Computation of Probabilities of Events Described by Order Statistics and Applications to Queue Inference," *Journal on Computing*, No.1, pp.89-100.