TELETRAFFIC SCIENCE for New Cost-Effective Systems,
Networks and Services, ITC-12
M. Bonatti (Editor)
Elsevier Science Publishers B.V. (North-Holland)

A QUEUEING NETWORK ANALYZER FOR MANUFACTURING

Moshe SEGAL[1] and Ward WHITT[2]

(1) AT&T Bell Laboratories, Room 3L-303, Holmdel, NJ 07733
(2) AT&T Bell Laboratories, Room 2C-178, Murray Hill, NJ 07974

We describe a new version of the Queueing Network Analyzer (QNA) software package that was developed especially to analyze manufacturing lines. The goal was to obtain a convenient tool for estimating capacity, work-in-process inventory and production intervals, as needed to design or change a manufacturing line. QNA is an analytic tool, based on mathematical formulas rather than simulation, and simple approximations rather than involved numerical procedures, so that it can produce results for relatively large and complex models quickly and inexpensively. In particular, QNA employs the parametric-decomposition approximation method, which has its roots in teletraffic theory. To meet needs in the manufacturing environment, QNA has been modified to represent machine breakdown, batch service, changing lot sizes and product testing with associated repair and partial yields. QNA also has a new menu-driven screen-oriented interface using manufacturing terminology.

1. INTRODUCTION

In 1983 a software package called the Queueing Network Analyzer (QNA) was developed at AT&T Bell Laboratories to calculate approximate performance measures for systems that can be modeled as general (non-Markov) open queueing networks [1], [2]. The model allows non-exponential service-time distributions, non-Poisson external arrival processes and deterministic routes through the network. The original purpose was to analyze packet communication networks; e.g., QNA is a principal component of the performance analysis module in a packet network design and analysis tool [3] and the QNA methodology has been used to analyze statistical multiplexers for voice and data [4]. However, as part of AT&T's increased effort to improve manufacturing operations [5], we have also been addressing applications in manufacturing. Our purpose here is to describe a new version of QNA developed especially for analyzing manufacturing lines.

1.1 A History in Teletraffic Theory

The early research on queueing networks, including the seminal work by J. R. Jackson [6], [7], was primarily motivated by manufacturing systems. This initial research eventually led to today's rich theory of product-form queueing networks, with recent progress stimulated by computer performance issues. To a large extent, the approximation methodology in QNA emerged from teletraffic theory. Overflow processes arising in telephone networks with alternate routing were soon found to be substantially more variable or "peaked" than a Poisson stream. The peakedness concept was developed to partially characterize overflow traffic, and the equivalent random method was developed to describe the resulting blocking (lost calls) [8], [9]. The performance of complicated telephone networks was analyzed approximately by what has come to be known as the parametric-decomposition method: The flows were partially characterized by one [10] and then two parameters [11], one to describe the rate and the other to describe the variability (peakedness), and the different nodes (trunk groups) were treated as independent given the two parameters partially characterizing the arriving traffic. The final parameters used to

represent the arriving traffic at each node were characterized by a system of equations, which could be solved iteratively.

1.2 The Same General Approach With Different Variability Parameters

This same parametric-decomposition approximation method is the basis for QNA and earlier efforts along the same lines [12], [13]. The previous teletraffic model and analysis do not apply directly, however, because now attention is focused on delays experienced at queues with waiting space instead of blocking at service facilities without waiting space. Of course, peakedness can be used to analyze delay systems [14], but it has become more common to partially characterize arrival processes in delay systems with the squared coefficient of variation (variance divided by the square of the mean) of an interarrival time, denoted by c_a^2. As a second parameter for delay systems, c_a^2 is not automatically better than the peakedness (for some comparisons, see Section III of [15]); the primary advantage of c_a^2 is that it is easier to understand and interpret. However, the simplicity of c_a^2 is somewhat misleading. We think of the approximating process being a renewal process with c_a^2 being the squared coefficient of variation of an interarrival time [16], [17]. When the actual arrival process is nearly renewal, it is usually appropriate to obtain the approximation by just letting c_a^2 be the squared coefficient of variation of an interarrival time, or an approximation of it. However, in many cases, the actual arrival process is not nearly renewal, i.e., the variability of the process is largely due to the dependence (e.g., correlations) among successive interarrival times instead of the variability in the individual interarrival-time distribution [4]. Then an appropriate approximating variability parameter c_a^2 should represent this dependence in the process as well as the squared coefficient of variation of one interarrival time. We can still think of the approximating process as a renewal process, but it is important to realize that *renewal-process approximations do not automatically mean ignoring the dependence among successive interarrival times in the actual arrival process.* Moreover, c_a^2 for a fixed arrival process might be chosen to depend on the traffic intensity of the queue to which it is offered, because the way the dependence in the fixed arrival process affects the performance of the queue depends on the traffic intensity in the queue.

1.3 Overview of the Algorithm

The parametric-decomposition approximation is implemented in QNA by treating each queue as a standard GI/G/m queue with m servers, unlimited waiting space, the first-come first-served discipline, and a renewal arrival process, partially characterized by the mean λ^{-1} and squared coefficient of variation c_a^2 of the general interarrival-time distribution and the mean τ and squared coefficient of variation c_s^2 of the service-time distribution. For example, a simple approximation for the expected steady-state waiting time before beginning service is

$$EW(\lambda, c_a^2, \tau, c_s^2, m) = \left[\frac{c_a^2 + c_s^2}{2}\right] EW(\mathrm{M/M/m}, \lambda, \tau), \tag{1}$$

where $EW(\mathrm{M/M/m}, \lambda, \tau)$ is the exact value for the M/M/m queue; (70) of [2]. Refinements of (1) and approximations for the entire waiting-time and queue-length distributions are described in [2], [18] [19].

Steady-state performance measures at each queue such as (1) are calculated in the middle of the algorithm. First, the parameters $(\lambda, c_a^2, \tau, c_s^2)$ for each queue must be extracted from the model input; afterwards, the production intervals (time from start to finish) for different products must be described. The flow through the network is specified in the model input of QNA primarily by *deterministic routes*, as described in Section 2.3 of [2]. We allow *multiple classes* (corresponding to different products). Associated with each class is a sequence of nodes (queues) called the *basic route*. For example, a possible basic route is the node sequence (3, 1, 7, 1, 2); the first node visit by this class is to node 3, the second node visit is to node 1, and so forth. This example illustrates that the node numbers can appear on the route in any order, that nodes can be visited more than once on the same route, and that some nodes need not be visited at all by that route. Associated with each route is a partial characterization of the external arrival process at the first node visit via an external arrival rate and an external arrival variability parameter. Also the service-time parameters τ and c_s^2 are specified for each node visit on each route. This allows different classes

to have different service-time distributions at the same node, and the same class to have different service-time distributions at different visits to the same node. The final arrival and service parameters $(\lambda, c_a^2, \tau, c_s^2)$ at each node are obtained by appropriately *aggregating* the input data above and solving *two systems of linear equations* that represent the three basic network operations of superposition (merging), splitting and flow through a queue (departure); see Sections 2.3 and 4 of [2]. The aggregation step yields the single-class model described in Section 2.1 of [2]. For the arrival rates, the system of linear equations is just the familiar *traffic rate equations*. For the quality of the approximations, it is significant that the resulting traffic intensity of each queue is *exact for this model*. For the variability parameters, the corresponding traffic variability equations are clearly an approximation but the approximation usually is reasonable. Difficulties with the variability parameters that can arise with aggregation are illustrated in [20].

To describe the production intervals for each class, we return to the basic routes. To describe the mean, we add the actual expected service times from the model input for that route plus the approximate expected waiting times at each node visited. To describe the approximate variance, we act as if the sojourn times (service time plus waiting time) at successive node visits are independent (an approximation), and add the actual variances of the service times $(\tau^2 c_s^2)$ as specified in the model input plus the variances of the waiting times at each node visit on the basic route. (Additional details appear toward the end of Section 2.)

2. NEW FEATURES FOR MANUFACTURING

We now describe features added to QNA since [2] in order to address needs in the manufacturing environment. Specific motivating applications are manufacturing lines for integrated circuits [21] and printed wiring boards. Models of wafer fabrication in integrated circuit manufacturing often have only a few products (1-10) and a modest number of workstations (5-50), but relatively long routes (30-300 node visits), because workstations are revisited to superimpose different layers on the wafers [21]. On the other hand, for printed wiring boards new flexible lines are being designed that have many different products (100-1000), each with their own sequence of operations (route) and processing characteristics (service-time distributions). QNA models of both kinds of lines are easy to build and analyze. Indeed, the primary task is obtaining meaningful model input data.

2.1 Partial Yields and Testing

With new technologies, the amount of defective product scrapped in the production process needs to be considered. Such *partial yields* are represented in QNA by incorporating probabilistic routing in the framework of deterministic routes. The partial yields are specified in the model input of QNA by having a probability of continuing on to the next node visit associated with each node visit on each route.

The basic routes are useful to represent most of the flow, but additional flexibility is needed to represent other possibilities such as test and repair. For example, after completing testing, 5% of some product might be junked and 10% might require rework. Moreover, some of the rework might be sent on place, while the rest might be sent some place else. To achieve the desired flexibility, *QNA allows additional probabilistic transitions, that are specified with reference to the basic routes.* The additional transitions are specified by eight parameters $(k_1, j_1; k_2, j_2; p, \gamma, \tau, c_s^2)$, which means that a transition from node visit j_1 on route k_1 to node visit j_2 on route k_2 occurs with probability p; given that the transition occurs, it is accompanied by a creation-or-combination factor γ; at the destination (node visit j_2 on route k_2) the service time distribution has mean τ and variability parameter c_s^2. The products experiencing this special transition thus do not necessarily have the same service-time distributions as the other class-k_2 customers at node visit j_2. Thereafter, however, this customer becomes a class-k_2 customer and follows that designated route (until, if ever, it experiences another special *hop transition*). As before [2], this additional flow and service specification, together with the basic route data, is converted into the single-class model with Markovian routing in Section 2.1 of [2] by aggregation;

the remaining analysis to calculate approximate performance measures at the queues is the same.

2.2 Changing Lot Sizes

To represent *changing lot sizes* (e.g., from chips to wafers containing many identical chips, to cassettes of wafers, to magazines of cassettes), QNA has a creation-or-combination factor associated with each node visit on each route. For example, a single departure might trigger a batch of seven arrivals at the next node visit (creation) or every seven consecutive departures might result in a single arrival at the next node visit (combination). The resulting change in arrival rate is obvious; the affect on variability is approximated as in Section 4.6 of [2], which is satisfactory to represent lot size changes that are not too great. Combination also means that we need to account for units waiting for the batch to form (discussed further below).

2.3 Batch Service

Batch service is incorporated into QNA by changing the lot size before and after service. Products are combined before service to form a batch and products are created after service to return to their original form. First, this means that we must keep track of units that are in the system waiting for batches to form; this leads to a three-part classification of delay: *batching time*, waiting time and service time. Second, it means that the basic performance measures at each node are for batches instead of individual units. When the batch size b is large, the batching time can be the dominant part of the total delay, so it is important to include it. The batching time is the sum of k interarrival times, with $k=0, 1, 2, ..., b-1$, each with probability $1/b$. The number of customers waiting for the batch to form has mean $(b-1)/2$, second moment $(b-1)(2b-1)/6$, variance $(b+1)(b-1)/12$ and squared coefficient of variation $(b+1)/3(b-1)$. If the unit interarrival times have mean λ^{-1} and squared coefficient of variation c_a^2, then the batching time has mean $\lambda^{-1}(b-1)/2$ and squared coefficient of variation $[6c_a^2+(b+1)]/3(b-1)$, by using the familiar formula for the variance of a random sum.

However, this analysis implicitly assumes that customers arrive as individual units. When customers arrive in batches, we calculate an approximate average input batch size I and an effective service batch size $E=\max\{1, b/I\}$. When $E \leqslant 1$, we approximate the number of customers batching by zero. This is exact when the service batch sizes of all queues feeding that queue are integral multiples of b, but not otherwise. When $E > 1$, we act as if the service batch size is E instead of b. Since the customers arrive in batches of size I, the number of customers waiting for a batch to form has a mean $I(E-1)/2=(b-I)/2$, second moment $I^2(E-1)(2E-1)/6$ and squared coefficient of variation $(E+1)/3(E-1)$. If the unit interarrival times have mean λ^{-1} and squared coefficient of variation c_a^2, then the resulting batching time has mean $I(E-1)/2\lambda$ and squared coefficient of variation $[6c_a^2I^{-1}+E+1]/3(E-1)$.

2.4 Service Interruptions

To represent *service interruptions* due to machine breakdowns, planned maintenance, etc., we include server availability parameters at each node. We initially model the availability of each server by an alternating renewal process; i.e., there is a succession of intervals $U_1, D_1, U_2, D_2, ...$ during which the server is alternately up (available to provide service) and down (unavailable to provide service). We assume that all these up and down times are mutually independent, for all nodes and all servers at each node, as well as within one sequence for one server. We also assume that all the up times for all servers at a node have a common distribution, partially characterized by its mean. Similarly, we assume that all down times for all the servers at a node have a common distribution, partially characterized by its mean and squared coefficient of variation. (We do not specify a variability parameter for the up time because it is not used in the algorithm.) Although the server availability parameters must be the same for all servers at the same node, they can vary from node to node.

We analyze this model approximately by acting as if the down times are triggered by service times. We assume that each product upon starting service causes a down time with probability p, and so has expanded service time equal to the original service time plus an independent down time, and has an ordinary service time with probability $1-p$. (When there is batch service, the down time is added to the service time of the batch.) We assume that successive down times are

generated independently (by independent trials, independent of the other processes). Note that this new model might actually be more realistic, e.g., when the down times are caused by products jamming machines. The modified model is much more tractable because it is again a standard *GI/G/m* queue, which can be analyzed using the same approximations. We first choose the down time probability p in order to produce the proper traffic intensity and, second, capture the principal effect of the increased variability caused by calculating a revised service-time variability parameter. The adjusted service-time variability parameter also affects the approximation for the departure process and thus other queues in a queueing network model. Since the k^{th} moment of a mixture is the mixture of the k^{th} moments, we can express the first two moments of the modified service-time distribution with parameters $\bar{\tau}$ and $\bar{c}_s^2$ as

$$\bar{\tau} = p(\tau+d) + (1-p)\tau = \tau + pd \tag{2}$$

and

$$\begin{aligned} \bar{\tau}^2(\bar{c}_s^2+1) &= p[c_s^2\tau^2 + c_d^2 d^2 + (\tau+d)^2] + (1-p)[c_s^2\tau^2+\tau^2] \\ &= (c_s^2+1)\tau^2 + p[c_d^2 d^2 + 2d\tau + d^2], \end{aligned} \tag{3}$$

where d is the mean and c_d^2 is the squared coefficient of variation of a down time. We choose p so that the new traffic intensity $\bar{\rho}=\lambda\bar{\tau}/m$ is appropriately related to the original traffic intensity $\rho=\lambda\tau/m$; i.e., if u is the mean up time, then we should have $\bar{\rho} = \rho + d/(d+u)$, from which we obtain $\bar{\tau}=m\bar{\rho}/\lambda$ and $p=m/\lambda(d+u)$. To get $\bar{c}_s^2$, we apply (3), replacing p by $\min\{p, 1\}$.

2.5 Deterministic and Random Routing

After the aggregation step, the model becomes a single-class network. *To better represent the routing, which is largely deterministic, we have modified the way the variability of the flow from one node to another is approximated.* If c_{di}^2 is the variability parameter of the overall departure process from node i, q_{ij} is the proportion of the departures from node i routed to node j, and we use Markovian routing, then the approximate variability parameter c_{ij}^2 for the flow going from node i to node j is

$$c_{ij}^2 = q_{ij}\, c_{di}^2 + 1 - q_{ij}; \tag{4}$$

see (36) of [2]. If the overall departure process were actually renewal and the routing were Markovian, then (4) would be exact. However, in many manufacturing applications the routing is primarily deterministic for each product. As observed by Bitran and Tirupati [22], (4) can be a poor approximation in manufacturing models with multiple classes, low variability and deterministic routing. If q_{ij} is small, then (4) makes c_{ij}^2 nearly 1, when it should be much less. Based on [22]-[24], to treat deterministic routing, QNA uses

$$c_{ij}^2 = q_{ij}\, c_{di}^2 + (1-q_{ij})\, q_{ij}\, c_{ai}^2 + (1-q_{ij})^2\, c_{ei}^2, \tag{5}$$

where c_{ai}^2 is the overall arrival variability parameter at node i and c_{ei}^2 is an average of the external arrival-process variability parameters. In particular, we let $c_{ei}^2 = \sum_k n_{ki}\hat{c}_k^2 / \sum_k n_{ki}$, where the sums are over all basic routes, $\hat{c}_k^2$ is the external-arrival-process variability parameter for individual units on route k and n_{ki} is the expected number of visits to node i by route k, counting feedback and losses on that route (but not transitions from one route to another). For examples with purely deterministic routing, (5) performs better than (4), but in succeeding to represent deterministic routing, (5) necessarily fails to represent the random routing captured by (4). We thus introduce a further modification, and use a convex combination of (4) and (5)

$$c_{ij}^2 = q_{ij}\, c_{di}^2 + (1-q_{ij})(\beta_{ij}\, q_{ij}\, c_{ai}^2 + 1 - (\beta_{ij}(1-(1-q_{ij})c_{ei}^2), \tag{6}$$

where β_{ij} is the proportion of all flow from i to j that is due to deterministic routing (as opposed to random routing). In particular, as a reasonable approximation, we define β_{ij} as the proportion of all flow from i to j that is determined by the basic routes as opposed to the additional transitions. Formula (6) leads to a change in the traffic variability equations in Section 4.2 of [2].

2.6 Summary Performance Measures

As in [2], *the last step is to calculate the approximate mean and variance of the production interval for each class.* The three components – total batching time, total waiting time and total service time – are also described. We actually describe these production intervals in three different ways, because there are different interpretations. First, production intervals are calculated for the basic routes. With this scheme, each product is assumed to make every node visit on the basic route exactly once; i.e., we ignore losses early on the route (partial yields), we ignore feedback on the route (rework), and we ignore extra transitions to and from other routes. The second calculation also applies to individual routes, but *does* count losses and feedback. (It does not count any additional transitions that go from one route to another, though.) Since we rarely want to count short times spent by defective products that are discarded early on the route, this second scheme often produces less useful production interval statistics, but the associated average WIP (work-in-process inventory) calculated by Little's Law is usually what we want. In the third calculation we count feedback but not losses. Since feedback is counted while losses are not, this scheme thus leads to longer production intervals; it is useful to describe the production intervals of good product (including rework). Finally, based on the second production interval calculation above, QNA prints out for each route *the total yield* and *the proportion yield.* The total yield is the departure rate in units from the final node visit on the basic route. The proportion yield is the total yield divided by the arrival rate. This proportion yield can exceed 1.0 if lot sizes change.

3. COMPARISONS WITH LARGE MANUFACTURING SIMULATION MODELS

In this section we describe comparisons made between QNA and two simulation models of manufacturing lines. These lines produce a single product, so that there is only one route in each case. However, one model has 67 workstations (nodes) and 135 operations (node visits) on the one route, while the other has 30 workstations and 108 operations. Moreover, there are several partial-yield and rework specifications (hops). In fact, all the features described except changing lot sizes are present in these models. Batch service occurs at some workstations and various server availability scenarios are considered. A summary of the results appears in Table 1.

Table 1. Comparisons Between QNA and Simulation

	Number of				WIP (Lots) Mean		Yield	Interval (Time Units) Mean		Interval (Time Units) Standard Deviation	
Model	Workstations	Products	Operations	Hops	QNA	SIM	QNA/SIM	QNA	SIM	QNA	SIM
1	67	1	135	9	261.6	255.5	1.0045	34.2	32.4	4.81	4.97
2	30	1	108	15	41.8	40.5	1.000	12.7	12.4	2.8	–

Since the QNA utilization and yield calculations are exact, they provided good preliminary checks that the two models are really the same. Of course, perfect agreement is not possible because the simulation is subject to statistical fluctuations. Of particular interest are the QNA estimates of average work-in-process inventory (WIP) and the average production interval. These results shown in Table 1 are very encouraging, suggesting that the QNA approximations are sufficiently accurate for QNA to be profitably employed to provide quick estimates of capacity, WIP and production intervals.

ACKNOWLEDGEMENTS

The QNA software was written by A. J. Cipolone, F. F. Reed, R. Sandt and A. T. Seery. The simulation models in Section 3 were built and run by F. J. Gurrola-Gal.

REFERENCES

[1] Whitt, W., Approximations for Networks of Queues, *Proc. Tenth Int. Teletraffic Congress*, Montreal, 1983, 4.1.2.

[2] Whitt, W., The Queueing Network Analyzer, *Bell System Tech. J.* 62 (1983) 2779-2815.

[3] Monma, C. L. and Sheng, D. D., Backbone Network Design and Performance Analysis: A Methodology for Packet-Switching Networks, *IEEE J. Select. Areas Commun.* SAC-4 (1986) 946-965.

[4] Sriram, K. and Whitt, W., Characterizing Superposition Arrival Processes in Packet Multiplexers for Voice and Data, *IEEE J. Select. Areas Commun.* SAC-4 (1986) 833-846.

[5] *Manufacturing for Technology, AT&T Tech. J.* 65 (July-August 1986).

[6] Jackson, J. R., Networks of Waiting Lines, *Operations Res.* 5 (1957) 518-521.

[7] Jackson, J. R., Jobshop-like Queueing Systems, *Management Sci.* 10 (1963) 131-142.

[8] Wilkinson, R. I., Theories of Toll Traffic Engineering in the U.S.A, *Bell System Tech. J.* 35 (1956) 421-514.

[9] Bretschneider, G., Die Berechnung von Leitungsgruppen fur uberfliessender Verkehr in Fernsprechwahlanlagen, *Nachrichtentechnische Zeitschrift* 9 (1956) 533-540.

[10] Segal, M., Traffic Engineering of Communications Networks with General Class of Routing Schemes, *Fourth Int. Teletraffic Congress*, London, 1964, No. 24.

[11] Katz, S., Statistical Performance Analysis of a Switched Communications Network, *Proc. Fifth Int. Teletraffic Congress*, New York, 1967, pp. 566-575.

[12] Reiser, M. and Kobayashi, H., Accuracy of the Diffusion Approximation for Some Queueing Systems, *IBM J. Res. Dev.* 18 (1974), 110-124.

[13] Kuehn, P. J., Approximate Analysis of General Queueing Networks by Decomposition, *IEEE Trans. Commun.* COM-27 (1979) 113-126.

[14] Heffes, H., Analysis of First-Come First-Served Queueing Systems with Peaked Inputs, *Bell System Tech. J.* 52 (1973) 1215-1228.

[15] Whitt, W., On Approximations for Queues, I: Extremal Distributions, *AT&T Bell Lab. Tech. J.* 63 (1984) 115-138.

[16] Whitt, W., Approximating a Point Process by a Renewal Process, I: Two Basic Methods, *Operations Res.* 30 (1982) 125-147.

[17] Albin, S. L., Approximating a Point Process by a Renewal Process, II: Superposition Arrival Processes to Queues, *Operations Res.* 32 (1984) 1133-1162.

[18] Kraemer, W. and Langenbach-Belz, M., Approximate Formulae for the Delay in the Queueing System GI/G/1, *Proc. Eighth Int. Teletraffic Congress*, Melbourne, 1976, 235-1/8.

[19] Whitt, W., Approximations for the GI/G/m Queue, *Adv. Appl. Prob.*, to appear.

[20] Fendick, K. W., Saksena, V. R. and Whitt, W., Dependence in Packet Queues: A Multi-Class Batch Poisson Model, this volume.

[21] Burman, D. Y., Gurrola-Gal, F. J., Nozari, A., Sathaye, S. and Sitarik, J. P., Performance Analysis Techniques for IC Manufacturing Lines, *AT&T Tech. J.* 65 (July-August 1986) 46-57.

[22] Bitran, G. and Tirupati, D., Multiproduct Queueing Networks with Deterministic Routing: Decomposition Approach and the Notion of Interference, *Management Sci.*, to appear.

[23] Whitt, W., Approximations for Single-Class Departure Processes from Multi-Class Queues, submitted.

[24] Whitt, W., A Light-Traffic Approximation for Single-Class Departure Processes from Multi-Class Queues, *Management Sci.*, to appear.