Distributed Sensor Networks, 2nd edition
Vol. 2, Section II.I, Chapter #47
Quality of Service Metrics with Applications to Sensor Networks

N. Gautam*
Texas A&M University,
235A Zachry, Mailstop 3131
College Station, TX 77843-3131
gautam@tamu.edu

*The author was partially supported by NSF grant CMMI–0946935

1 Service Systems

The phrase "Quality of Service" (QoS) has been popular for over 20 years, however there has been little or no consensus in terms of what QoS actually is, what various QoS metrics are, and what QoS specifications are. Yet, QoS has spread far and wide, beyond the realm of networking (where the term QoS was first used), into areas such as transportation, health care, hospitality, manufacturing, etc. In fact, the author believes it may be better to introduce QoS using examples from the service industry to provide appropriate analogies in the hope of giving the study of QoS more structure as well as discover newer ways of providing QoS in computer networks and then finally get more specific to discuss QoS in distributed sensor networks. To define a service industry, one must first differentiate between goods which are usually tangible, and services which typically are intangible. In fact several organizations that have been traditionally concentrating on their goods (such as cars at vehicle manufacturers, food at restaurants, books at bookstores, etc.) are now paying a lot of attention to service (such as on-time delivery, availability, warranties, return policies, overall experience, etc.). These typically fall under the realm of providing QoS.

1.1 Elements of a Service System

Examples of service systems range from complex inter-connected networks such as computer-communication networks, transportation systems, theme parks, etc. to simpler individual units such as a barber shop, repair shops, theaters, restaurants, hospitals, hotels, etc. In all these examples two key players emerge, namely, the service provider and users. As the names suggest, the users receive service provided by the service provider. Users (also called customers if there is money involved) do not necessarily have to be humans, they could be other living or non-living entities. Further, users do not have to be single individuals, they could be part of a group (such as in a multicast session, in a restaurant, at a play, etc.). On the same token, for a given system, there could be zero, one or many service providers. Although most services are such that they are owned by a single entity (the one to blame if things go wrong), there are some (including the Internet) that are owned by several groups.

QoS can be defined as a set of measures that the users "want" from the system (or sometimes what the service provider wants to give the users). What the users eventually "get" is called performance. From a physical goods standpoint, QoS is equivalent to specifications (or specs as they are usually abbreviated). Some QoS measures are qualitative (such as taste, ambiance, etc.) and these are hard to provide since different users perceive them differently. Other QoS measures which are quantitative also have some fuzziness attached. For example on one day a user might find a 90ms latency intolerable and on another day the user may find 100ms latency tolerable. There could be several reasons for that including the mood of the user, the expectations of the user, etc. Capturing such cognitive aspects are beyond the scope of this chapter. We will focus on systems where user requirements (i.e. QoS) are known precisely and users are satisfied or unsatisfied if the requirements are met or not met respectively. That means if 100ms is the tolerance for latency, then QoS is met (not met) if latency is lesser (greater) than 100ms.

In some service systems the users and the service providers negotiate to come up with what is known as a *Service Level Agreement* (SLA). For example, years ago a pizza-company promised to deliver pizzas within 45 minutes, or the pizzas are free. That is an example of an SLA which is also called a QoS guarantee. In many service systems there isn't an explicit guarantee, but a QoS indication such as: "your call will be answered in about 3 minutes", "the chances of a successful surgery is 99.9%", "the number of defective parts is in the order of one in a million", etc. In many systems it is not possible to provide absolute QoS guarantees due to the dynamic nature of the system, but it may be feasible to deliver relative QoS. This is typically known as *Level of Service* (LoS) where for example if there are 3 LoS's, gold, silver and bronze, then at a given time instant,

gold level will get better QoS than silver level which would get a better QoS than bronze level.

1.2 Customer Satisfaction

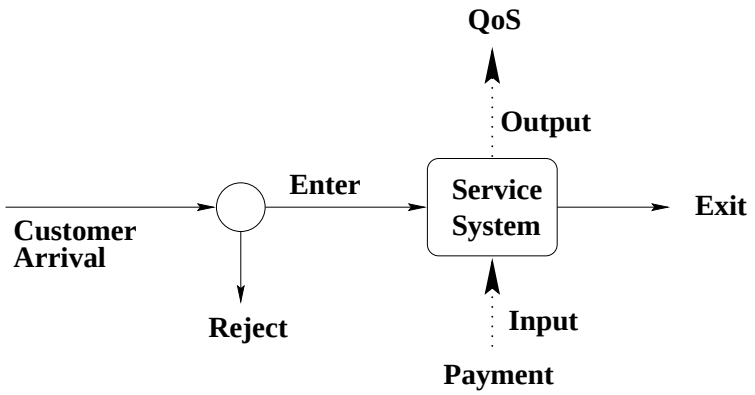

Figure 1: Customer satisfaction in a service system

Although many consider QoS and customer satisfaction as one and the same, here QoS is thought of as only a part of customer satisfaction. However it is not assumed here that providing QoS is the objective of a service provider, but providing customer satisfaction is. With that understanding, the three components of customer satisfaction are: (a) QoS, (b) availability, and (c) cost. The service system (with its limited resources) can be considered either physically or logically as one where customers arrive, if resources are available they enter the system, obtain service for which the customers incur a cost, and then they leave the system (see Figure 1). One definition of availability is the fraction of time arriving customers enter the system. Thereby QoS is provided only for customers that "entered" the system. From an individual customer's standpoint, the customer (i.e. user or application) is satisfied if the customer's requirements over time on QoS, availability and cost are satisfied. Some service providers' objective is to provide satisfaction aggregated over all customers (as opposed to providing absolute service to an individual customer). Services such as restaurants provide both, they cater to their frequent customers on one hand and on the other hand they provide overall satisfaction to all their customers.

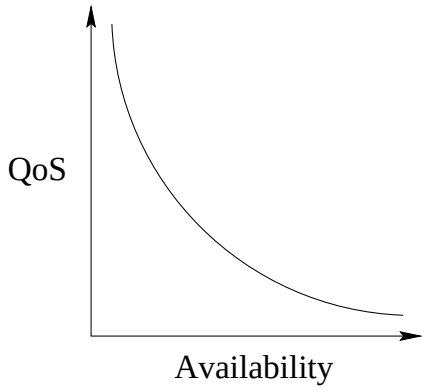

Figure 2: Relationship between QoS and availability, given cost

The issue of QoS (sometimes called conditional QoS as the QoS is conditioned upon the ability to enter the system) versus availability needs further discussion. Consider the analogy of visiting a medical doctor. The ability to get an appointment translates to availability, however once an

appointment is obtained, QoS pertains to the service rendered at the clinic such as waiting time, experience, healing time, etc. Another analogy is airline travel. Getting a ticket on an airline at a desired time from desired source to desired destination is availability. QoS measures include delay, smoothness of flight, in-flight service, etc. One of the most critical business decisions is to find the right balance between availability and QoS. The two are inversely related as illustrated in Figure 2. A service provider can increase availability by decreasing QoS and vice versa. A major factor that could affect QoS and availability is cost. Usually with cost (somewhat related to pricing) there is a need to segregate the customers into multiple classes. It is not necessary that classes are based on cost, they could also depend on customer type (i.e. applications) and QoS requirements. The ability to provide appropriate customer satisfaction based on class (and relative to other classes) is a challenging problem especially under conditions of stress, congestion, unexpected events, etc. For example if an airplane encounters turbulence, all customers experience the bumpy ride, irrespective of the class of service.

1.3 Effect of Resources and Demand

Customer satisfaction is closely related to both resources available at the service provider as well as demand from the customers. It is of grave importance to understand the relationship and predict or estimate it. Firstly consider the relationship between resource and performance. The graph of resources available at a service provider versus the performance the service provider can offer is usually as described in Figure 3. From Figure 3 the following are evident: (1) it is practically impossible to get extremely high performance, and (2) to get a small increase in performance it would sometimes even require twice the amount of resources, especially when the available performance is fairly high in the first place.

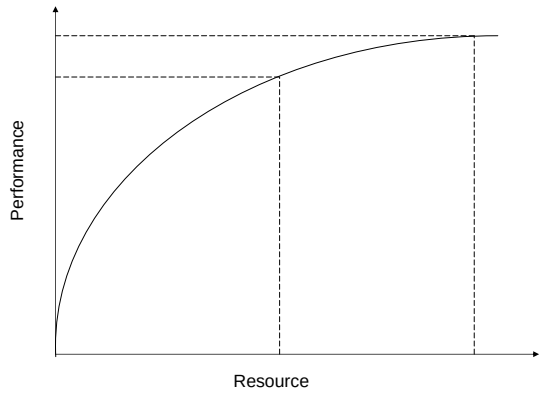

Figure 3: Resource versus performance

Now consider the relationship between customer satisfaction and demand. If a service offers excellent customer satisfaction, very soon it's demand would increase. However if the demand increases, the service provider would no longer be able to provide the high customer satisfaction which eventually deteriorates. Thereby demand decreases. This cycle continues until one of two things happen, either the system reaches an equilibrium or the service provider goes bankrupt. The situations are depicted in Figure 4. Notice the relation between customer satisfaction and

demand, they are inversely related from the service provider's standpoint and directly related from a customer's standpoint.

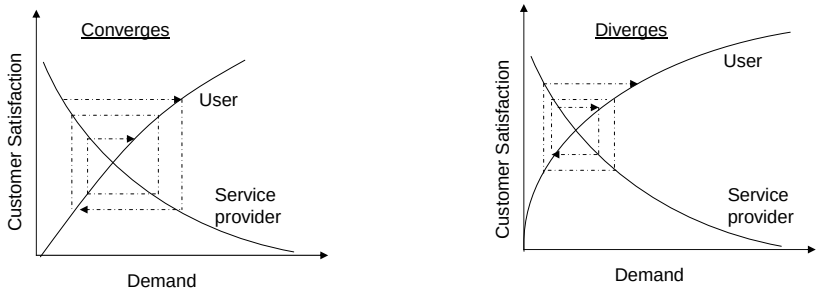

Figure 4: Demand versus customer satisfaction

After studying some general aspects of providing QoS in the service industry, we now turn our attention to QoS provisioning in networking in the next section.

2 QoS in Networking

Although the Internet started as a free and best-effort service, the next generation Internet is showing signs of becoming a network based on both pricing issues as well as QoS. The concept, need and application of QoS has also quickly spread to other high-speed networks such as telephony, peer-to-peer networks, sensor networks, ad-hoc networks, private networks, etc. For the remainder of this chapter on QoS, we will use networking rather loosely in the context of any of the above networks. We will not be restricting ourselves to any particular network or protocols running on the network. It may be best to think of an abstract network with nodes (or vertices) and arcs (or links). In fact the nodes and arcs can be either physical or logical in reality. At each node there are one or more queues and processors that forward information. With this setting in mind, we proceed to investigate the broad topic of QoS in high-speed networks.

2.1 Introduction

The plain old wired telephone network, one of the oldest computer-communication networks is one of the only high-speed networks that can provide practically perfect QoS, at least in the United States. One of the positive aspects of this is that give other networks (such as the Internet) time and they would eventually become as effective as the telephone networks in terms of providing

QoS. Having said that, it is important to notice that under some cases the telephone network does struggle to provide QoS; one example is during the terrorist attacks on September 11, 2001 – it was impossible to dial into several areas; another example is international phone calls, there is a long way to perfecting QoS for that. With the advent of cell phones, the telephone networks are now facing a new challenge of providing QoS to wireless customers.

From a networking standpoint (for Internet-type networks), one of the difficulties for providing QoS is the presence of multiple classes of traffic such as voice, video, multimedia, data, web, etc. Unlike wired telephone networks that offer a bandwidth of about 60kbps for whatever type of call (regular phone calls, fax, modem call, etc.), in networking various types of traffic have varying requirements. In fact real-time traffic can tolerate some loss but very little delay, however non-real-time traffic cannot tolerate loss but can take a reasonable amount of delay. The number of network applications are increasing steadily, however it is not practical to support more than a handful of classes (2-4 classes). Therefore clever traffic aggregation schemes need to be developed.

In order to provide QoS in a multi-class network some of the important aspects to consider and optimize are: scheduling, switching and routing. For differentiating the various classes of traffic as well as providing different QoS, information needs to be processed in a manner other than first-come-first-served (FCFS). The essence of scheduling is to determine what information to serve next. The telephone network which essentially has only one class of traffic does not do any special scheduling. It just does switching and routing. Switching is done using circuit switching policies where upon dialing a number, a virtual path is created from the source to the destinations through which information is transmitted. An appropriate routing algorithm is used to efficiently send information from the source to the destination. From a networking standpoint, doing appropriate scheduling, switching and routing it would be possible to provide QoS to the users. How to do that is being pursued actively by the research community.

2.2 Characteristics of Network QoS Metrics

Before looking at how to provision QoS in networking applications, it is important to understand what the QoS metrics are in the first place and how to characterize them. There are four main QoS metrics in networking: delay, jitter, loss and bandwidth. There are other derived metrics used in sensor networks but they are all a function of some or all of the main QoS metrics.

- *Delay*: It is defined as the time elapsed between when a node leaves a source and reaches a destination. Though the term delay implies there is a target time and the information comes after the target time elapses, that really is not the case. It is just a measure of travel time from source to destination which is also called latency or response time.

- *Jitter*: The variation in the delay is termed as jitter. If a stream of packets are sent from a source to a destination, typically all packets do not face the same delay. Some packets experience high delays and others experience low delays. Applications such as video-on-demand can tolerate delays but not jitter. A simple way of eliminating or reducing jitter is to employ a jitter buffer at the destination. All packets are collected and then transmitted. This does increase the delay though. For that reason it is not common to see jitter, instead most articles focus on delay.

- *Loss*: When a piece of information arrives at a node at a time when the queue at the node is full (i.e. full buffer) or the node is not available for other reasons, then the information is dropped (or lost). This is known as loss. There are several measures of loss including loss probability (the probability that a piece of information can be lost along its way from its source to its destination) and loss rate (the average amount of information lost per unit time in a network or node).

- *Bandwidth*: Several real-time applications such as voice over IP, video-on-demand, etc. require a certain bandwidth (in terms of bytes per second) to be available for successful transmission. In fact the only QoS guarantee a telephone network provides is bandwidth (of about 60kbps).

Note: As alluded to in Section 1.2, while studying QoS, the concept of availability is skipped, however it is very important from a customer-satisfaction standpoint to consider availability.

Performance metrics can be typically classified into three parts: additive, multiplicative and minimal. In order to explain them, consider a traffic stream that originates at a particular node, and traverses through N nodes before reaching its destination. The objective is to obtain end-to-end performance metrics given metrics across nodes. For example consider node i (for $i \in [1, N]$), let d_i, ℓ_i and b_i respectively be the delay, loss probability and bandwidth across node i. Assume that the performance metrics across node i are independent of all other nodes. To compute end-to-end performance measures the following are used –

- *Additive*: The end-to-end performance measure is the sum of the performance measures over the individual nodes along the path or route. The end to end delay (D) for our above example is obtained as
$$D = d_1 + d_2 + \ldots + d_N = \sum_{1=1}^{N} d_i.$$
- *Multiplicative*: The end-to-end performance measure is the product of the performance measures over the individual nodes along the path or route. The end to end loss (L) for our above example is obtained as
$$L = 1 - (1 - \ell_1)(1 - \ell_2) \ldots (1 - \ell_N).$$
Note that the multiplicative metric can be treated as an additive metric by taking the logarithm of the performance measure.
- *Minimal*: The end-to-end performance measure is the minimum of the performance measures over the individual nodes along the path or route. The end to end bandwidth (B) for our above example is obtained as the minimum bandwidth available across all the nodes in its path. In particular,
$$B = \min\{b_1, b_2, \ldots, b_N\}.$$

Although all performance metrics are inherently stochastic and time-varying, in order to keep analysis tractable the following are typically done: replace a metric by its long-run or steady state or stationary equivalent; use an appropriate deterministic value such as maximum or minimum or mean or median or mode; use a range of meaningful values. Now, in order to guarantee QoS, depending on whether a deterministic or stochastic performance metric is used, the guarantees are going to be either absolute or probabilistic respectively. For example you could give an absolute guarantee that the mean delay is going to be less than 100ms. Or you could say that the probability that the delay is greater than 200ms is less than 5%. It is also possible to get bounds on the performance and it is important to note that giving deterministic bounds could mean under-utilization of resources and thereby very poor availability. Once again, it is crucial to realize that for a given infrastructure, the better QoS guarantees one can provide, the worse off will be availability (see Figure 2).

3 Systems Approach to QoS Provisioning

In this section we focus on obtaining performance metrics which form the backbone of QoS analysis. There are several methodologies to evaluate the performance of a system and they can be broadly

classified into experimental, simulation-based and analytical methods. Experimental methods tend to be expensive and time-consuming, whereas require very little approximations. The analytical models are just the opposite. Simulation-based techniques fall in the middle of the spectrum. This chapter focuses on obtaining analytical results which would be mainly used in making optimal design and admission control decisions. These can be appropriately used for strategic, tactical and operational decisions depending on the time-granularity. In this section we present two main performance analysis tools based on queueing theory and large deviations theory.

3.1 Performance Analysis Using Queueing Models

We begin this section by considering a single station queue and then extend the theory to a network of queues. From an analysis standpoint the most fundamental queueing system is the $M/M/1$ queue. Input to the queue is according to a Poisson process with average rate λ per unit time (i.e. inter-arrival times exponentially distributed). Service times are exponentially distributed with mean $1/\mu$. It is a single server queue with infinite waiting room and FCFS service. The following performance measures can be derived (under the assumption $\lambda < \mu$): average number in the system is $\frac{\lambda}{\mu-\lambda}$ and average waiting time in the system is $\frac{1}{\mu-\lambda}$. For the $M/M/1$ queue, distribution of the waiting times is given by

$$P\{\text{Waiting Time } \leq x\} = 1 - e^{-(\mu-\lambda)x}.$$

In this way other generalizations to this model such as a different arrival process, or a different service time distribution, or more number of servers, finite waiting room, different order of service, etc. can be studied. The reader is referred to one of several standard texts on queues (such as Gross and Harris [10], Bolch et al [1]).

Now we turn to a network of queues, specifically what is known as a *Jackson Network*. The network consists of N service stations (or nodes). There are s_i servers at node i. Service times at node i are exponentially distributed and independent of those at other nodes. Each node has infinite waiting room. Externally customers arrive at node i according to a Poisson process with mean rate θ_i. Upon completion of service at node i a customer departs the system with probability r_i or joins the queue at node j with probability p_{ij}. Assume that at least one node has arrivals externally and at least one node has customers departing the system. The vector of effective arrival rates $\lambda = (\lambda_1\ \lambda_2\ \ldots\ \lambda_N)$ can be obtained using

$$\lambda = \theta(I - P)^{-1},$$

where $\theta = (\theta_1\ \theta_2\ \ldots\ \theta_N)$, P is the routing probability matrix composed of various $[p_{ij}]$ values and I is an $N \times N$ identity matrix. Then each queue i can be modeled as independent single station queues. Jackson's theorem states that the steady state probability of the network can be expressed as the product of the state probabilities of the individual nodes. An application of Jackson networks is illustrated in Section 4.1.

3.2 Performance Analysis Using Large Deviations Theory

In this section we will focus on using the principle of large deviations for performance analysis of networks based on fluid models. Although large deviations does not require fluid traffic, the reason we pay attention to it is that fluid models represent correlated and long-range dependent traffic very well. In fact the simple on-off source could be thought of as one that generates a set of packets back to back when it is on and nothing flows when it is off.

Let $A(t)$ be the total amount of traffic (fluid or discrete) generated by a source (or flowing through a pipe) over time $(0, t]$. For the following analysis consider a fluid model. Note that it is straightforward to perform similar analysis for discrete models as well. However the results will be

identical. Consider a stochastic process $\{Z(t), t \geq 0\}$ that models the traffic flow. Also let $r(Z(t))$ be the rate at which the traffic flows at time t. Then

$$A(t) = \int_0^t r(Z(u))du.$$

The *asymptotic log moment generating function* (ALMGF) of the traffic is defined as

$$h(v) = \lim_{t\to\infty} \frac{1}{t} \log E\{\exp(vA(t))\}.$$

Using the above equation, it is possible to show that $h(v)$ is an increasing, convex function of v and for all $v > 0$,

$$r^{mean} \leq h'(v) \leq r^{peak},$$

where $r^{mean} = E(r(Z(\infty)))$ is the mean traffic flow rate, $r^{peak} = \sup_z\{r(z)\}$ is the peak traffic flow rate, and $h'(v)$ denotes the derivative of $h(v)$ with respect to v. The *Effective Bandwidth* of the traffic is defined as

$$eb(v) = \lim_{t\to\infty} \frac{1}{vt} \log E\{\exp(vA(t))\} = h(v)/v.$$

It can be shown that $eb(v)$ is an increasing function of v and

$$r^{mean} \leq eb(v) \leq r^{peak}.$$

Also,

$$\lim_{v\to 0} eb(v) = r^{mean} \quad \text{and} \quad \lim_{v\to\infty} eb(v) = r^{peak}.$$

It is not easy to calculate effective bandwidths using the formula provided above. However, when $\{Z(t), t \geq 0\}$ is a Continuous Time Markov Chain (see Elwalid and Mitra [6], and Kesidis et al [12]), or a Semi-Markov Process (Kulkarni [15]), one can compute the effective bandwidths more easily. Also, see Krishnan et al [14] for the calculation of effective bandwidths for traffic modeled by fractional a Brownian motion.

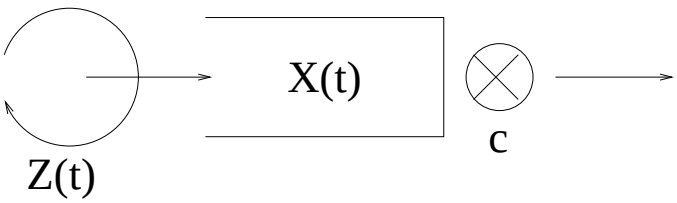

Figure 5: Single Buffer Fluid Model

Consider a single buffer fluid model as depicted in Figure 5. Input to a buffer of size B is driven by a random environment process $\{Z(t), t \geq 0\}$. When the environment is in state $Z(t)$, fluid enters the buffer at rate $r(Z(t))$. The output capacity is c. Let $X(t)$ be the amount of fluid in the buffer at time t. We are interested in the limiting distribution of $X(t)$, i.e.

$$\lim_{t\to\infty} P\{X(t) > x\} = P\{X > x\}.$$

Assume that the buffer size is infinite. In reality, the buffer overflows (hence packets/cells are lost) whenever $X(t) = B$ and $r(Z(t)) > c$. Note that the buffer content process $\{X(t), t \geq 0\}$ (when $B = \infty$) is stable if the mean traffic arrival rate is less than c, i.e.

$$E\{r(Z(\infty))\} < c.$$

Then $X(t)$ has a limiting distribution. From the limiting distribution, use $P\{X > B\}$ as an upper bound for loss probability (remember that B is the actual buffer size). This can also be used for delay QoS. Fluid arriving at time t waits in the buffer (hence faces a delay) for $X(t)/c$ amount of time. Therefore the long-run probability that the delay across the buffer is greater than δ is

$$P\{\text{Delay } > \delta\} = P\{X > c\delta\}.$$

Using results from large deviations, it is possible to show that for large values of x (specifically as $x \to \infty$),

$$P\{X > x\} \approx e^{-\eta x}$$

where η is the solution to

$$eb(\eta) = c.$$

Note that the above expression is an approximation, and in fact researchers have developed better approximations (see Elwalid and Mitra [7]) and bounds (see Gautam et al [9]). In fact Elwalid and Mitra [6] derive exact expressions for $P\{X > x\}$ for any CTMC environment $\{Z(t), t \geq 0\}$ process. We use these results and extensions in an example in Section 4.2.

To extend the single node results to a network of nodes, we need two important results. They are summarized as follows:

- *Effective Bandwidth of Output*: Refer to Figure 5. Let $D(t)$ be the total output from the buffer over $(0, t]$. The ALMGF of the output is

  $$h_D(v) = \lim_{t\to\infty} \frac{1}{t} \log E\{\exp(vD(t))\}.$$

  The effective bandwidth of the output traffic from the buffer is

  $$eb_D(v) = \lim_{t\to\infty} \frac{1}{vt} \log E\{\exp(vD(t))\}$$

  Let the effective bandwidth of the input traffic be $eb_A(v)$. Then the effective bandwidth $eb_D(v)$ of the output can be written as

  $$eb_D(v) = \begin{cases} eb_A(v) & \text{if } 0 \leq v \leq v^* \\ c - \frac{v^*}{v}\{c - eb_A(v^*)\} & \text{if } v > v^* \end{cases}$$

  where v^* is obtained by solving for v in the equation,

  $$\frac{d}{dv}\left[h_A(v)\right] = c.$$

  For more details refer to Chang and Thomas [2], Chang and Zajic [3] and de Veciana et al [4].

- *Multiplexing Independent Sources*: Consider a single buffer that admits a single-class traffic from K independent sources. Each source k ($k = 1, \ldots, K$) is driven by a random environment process $\{Z^k(t), t \geq 0\}$. When source k is in state $Z^k(t)$, it generates fluid at rate $r^k(Z^k(t))$ into the buffer. Let $eb_k(v)$ be the effective bandwidths of source k such that

  $$eb_k(v) = \lim_{t\to\infty} \frac{1}{vt} \log E\{\exp(vA_k(t))\}$$

  where

  $$A_k(t) = \int_0^t r^k(Z^k(u))du.$$

Let η be the solution to

$$\sum_{k=1}^{K} eb_k(\eta) = c.$$

Notice that the effective bandwidth of independent sources multiplexed together is the sum of the effective bandwidths of the individual sources. The effective bandwidth approximation for large values of x yields

$$P\{X > x\} \approx e^{-\eta x}.$$

4 Case Studies

In this section, case studies are presented to illustrate various performance analysis methodologies as well as to obtain various QoS metrics. The examples are kept simple purely for the purpose of illustration.

4.1 Case 1: Delay and Jitter QoS Metrics Using Queueing Networks

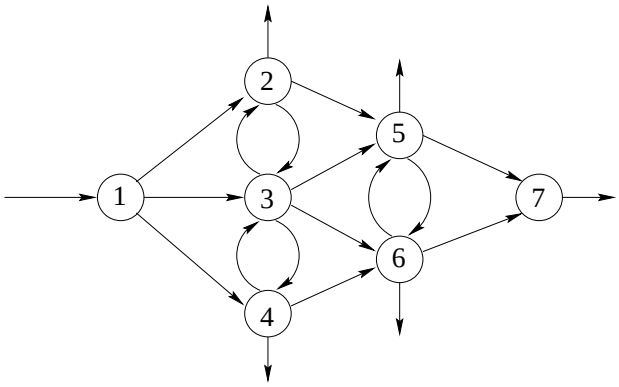

Figure 6: Server Configuration

Problem 1 *Consider a system of servers arranged as shown in Figure 6. Assume that requests arrive according to a Poisson process and enter node 1 at an average rate of 360 per minute. These requests can exit the system from nodes 2, 4, 5, 6 or 7. The processing time for each request in node j (for $j = 1, \ldots, 7$) is exponentially distributed with mean $1/\mu_j$ minutes. The vector of processing rates assume the following numerical values $[\mu_1\ \mu_2\ \mu_3\ \mu_4\ \mu_5\ \mu_6\ \mu_7] = [400\ 200\ 300\ 200\ 200\ 200\ 150]$. There is a single server at each node. When processing is complete at a node, the request leaves through one of the out-going arcs (assume the arcs are chosen with equal probability). The objective is to obtain the average delay and jitter experienced by all requests. In addition the average delay and jitter experienced by a particular arriving request that goes through nodes 1–2–5–7 and then exits the network is to be determined. Note that in this example, jitter is defined as the standard deviation of delay.*

Solution. The system can be modeled as a Jackson network with $N = 7$ nodes or stations. The external arrival rate vector $\theta = [\theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_6\ \theta_7]$ is

$$\theta = [360\ 0\ 0\ 0\ 0\ 0\ 0].$$

The routing probabilities are

$$P = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 \\ 0 & 1/4 & 0 & 1/4 & 1/4 & 1/4 & 0 \\ 0 & 0 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The effective arrival rate into the seven nodes can be calculated using $\theta(I-P)^{-1}$ as

$$[\lambda_1\ \lambda_2\ \lambda_3\ \lambda_4\ \lambda_5\ \lambda_6\ \lambda_7] = [360\ 180\ 240\ 180\ 180\ 180\ 120].$$

Now, each of the 7 nodes can be considered as independent $M/M/1$ queues due to Jackson's theorem. Let L_i be the number of requests in node i in the long run. For $i = 1, 2, \ldots, 7$, the mean and variance of the number of requests in node i are

$$E[L_i] = \frac{\lambda_i}{\mu_i - \lambda_i}$$

and

$$Var[L_i] = \frac{\lambda_i \mu_i}{(\mu_i - \lambda_i)^2}$$

respectively. Plugging in the numerical values, the average number of pending requests in the seven nodes can be computed as $[E[L_1]\ E[L_2]\ E[L_3]\ E[L_4]\ E[L_5]\ E[L_6]\ E[L_7]] = [9\ 9\ 4\ 9\ 9\ 9\ 4]$. Likewise the variance of the number of pending requests in the seven nodes can be computed as $[Var[L_1]\ Var[L_2]\ Var[L_3]\ Var[L_4]\ Var[L_5]\ Var[L_6]\ Var[L_7]] = [90\ 90\ 20\ 90\ 90\ 90\ 20]$. Let L be the total number of requests in the system of 7 nodes in the long run. Due to the fact that $L = L_1 + \ldots + L_7$ and independence between nodes, we have

$$E[L] = \sum_{i=1}^{7} E[L_i] = 53$$

and

$$Var[L] = \sum_{i=1}^{7} Var[L_i] = 490.$$

Let W be the time spent in the network by a request. The performance metrics of interest, namely delay and jitter are $E[W]$ and $\sqrt{Var[W]}$ respectively. Using Little's formula and its extensions (see Gross and Harris [10]) we have

$$E[W] = \frac{E[L]}{\sum_i \theta_i}$$

and

$$Var[W] = \frac{Var[L] + \{E[L]\}^2 - E[L]}{\left(\sum_i \theta_i\right)^2} - \{E[W]\}^2.$$

Therefore, the average delay and jitter experienced by all requests are 0.1472 minutes and 0.0581 minutes respectively.

Now, in order to determine the average delay and jitter experienced by a particular arriving request that goes through nodes 1–2–5–7, we use the fact that the time spent by a request

in node i is exponentially distributed with parameter $(\mu_i - \lambda_i)$. Therefore the mean and variance respectively of the time spent in nodes 1, 2, 5 and 7 are [0.0250 0.0500 0.0500 0.0333] and [0.0006 0.0025 0.0025 0.0011]. Since the nodes are independent, the mean and variance of the total times are the sum of those spent at the individual nodes. Therefore the average delay and jitter experienced by a particular arriving request that goes through nodes 1–2–5–7 are 0.1583 minutes and 0.0821 minutes respectively. ■

4.2 Case 2: Loss QoS Metrics Using Fluid Models

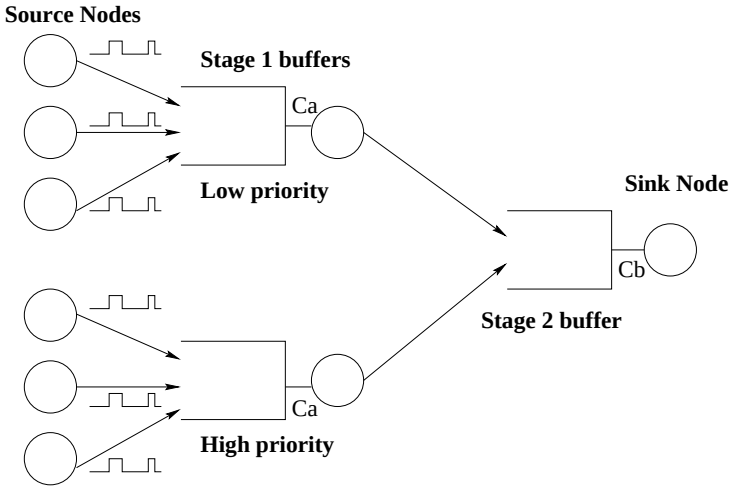

Figure 7: Sensor Network Configuration

Problem 2 *Consider a centralized sensor network as shown in Figure 7. The six source nodes send sensor data to a sink that processes the data. The intermediary nodes are responsible for multiplexing and forwarding information. The top three sources generate low priority traffic and the bottom three generate high priority. Subscript 1 is used for high priority and subscript 0 for low priority. The sensor data traffic can be modeled using on-off sources such that when the source is on, data is being generated at rate r_i for priority i and when it is off, no data is generated. Let the on and off times be exponentially distributed with parameters α_i and β_i respectively for priority i. Let c_a and c_b be the channel capacity of the buffers as shown in Figure 7. All buffers are of size B. However, assume that the second stage buffer is partitioned such that a maximum of B_i amount of priority i traffic can be stored in the buffer of size $B = B_0 + B_1$. In addition, at the second stage buffer, the high priority traffic is given all available/required processing capacity, anything remaining is given to the low priority traffic. For this system it is to be determined what loss probability QoS requirements can be met for both priorities. We use the following numerical values:*
$\alpha_0 = 1$, $\alpha_1 = 2$, $\beta_0 = 0.2$, $\beta_1 = 0.1$, $r_0 = 1$, $r_1 = 2.8$, $c_a = 1.5$, $c_b = 1.35$, $B_0 = 9$, $B_1 = 3$ *and* $B = B_0 + B_1 = 12$.

Solution. First consider the stage 1 buffers. The two buffers are identical in all respects except that the two buffers serve different classes of traffic. The main difference is in the subscript. So we perform the analysis using a subscript i to denote the priority. Let $eb_{i1}(v)$ be the effective bandwidth of the i^{th} priority traffic into the corresponding stage 1 buffer. The sources are exponential on-off sources, i.e., they can be modeled as a CTMC. Therefore we can use Elwalid and Mitra [6] to

obtain the effective bandwidth. Since the effective bandwidth of 3 independent sources multiplexed together is the sum of the effective bandwidths, we have

$$eb_{i1}(v) = \frac{3}{2v}\left(r_i v - \alpha_i - \beta_i + \sqrt{(r_i v - \alpha_i - \beta_i)^2 + 4\beta_i r_i v}\right).$$

Further, since the source is a CTMC, we can use exact analysis (from Elwalid and Mitra [5]) as opposed to the large deviations result. The loss probability for priority i traffic at stage 1 buffer (ℓ_{i1}) is given by

$$\ell_{i1} = \frac{3\beta_i r_i}{c_a(\alpha_i + \beta_i)} e^{-\eta_{i1} B}$$

where η_{i1} is the solution to $eb_{i1}(\eta_{i1}) = c_a$ which yields

$$\eta_{i1} = 3\alpha_i/(3r_i - c_a) - 3\beta_i/c_a.$$

Notice that if we had used large deviations, ℓ_{i1} would have been just $e^{-\eta_{i1} B}$ without the constant in front. Plugging in the numerical values (for the exact result, not large deviations), we get $\ell_{01} = 4.59 \times 10^{-9}$ and $\ell_{11} = 3.24 \times 10^{-4}$.

Now consider the stage 2 buffers. In order to differentiate the traffic, we will continue to use subscript i to denote the priority. Let $eb_{i2}(v)$ be the effective bandwidth of the i^{th} priority traffic into the stage 2 buffer. Using the result for effective bandwidth for the output from buffer of stage 1, we have

$$eb_{i2}(v) = \begin{cases} eb_{i1}(v) & \text{if } 0 \le v \le v_i^* \\ c_a - \frac{v_i^*}{v}\{c_a - eb_{i1}(v_i^*)\} & \text{if } v > v_i^* \end{cases}$$

with

$$v_i^* = \frac{\beta_i}{r_i}\left(\sqrt{\frac{c_a \alpha_i}{\beta_i(3r_i - c_a)}} - 1\right) + \frac{\alpha_i}{r_i}\left(1 - \sqrt{\frac{\beta_i(3r_i - c_a)}{c_a \alpha_i}}\right).$$

For the numerical values above, we get $v_0^* = 0.8$ and $v_1^* = 0.4105$. The loss probability for priority i traffic at stage 2 buffer (ℓ_{i2}) is given by

$$\ell_{i2} = e^{-\eta_{i2} B}$$

where η_{12} is the solution to $eb_{12}(\eta_{12}) = c_b$ and η_{02} is the solution to $eb_{02}(\eta_{02}) + eb_{12}(\eta_{02}) = c_b$. This is based on the results in Elwalid and Mitra [7] and Kulkarni and Gautam [16]. Plugging in the numerical values, we get $\ell_{02} = 0.0423$ and $\ell_{12} = 0.003$.

Assuming that the loss probability are independent across the two stages, the loss QoS requirements that can be satisfied are $1 - (1 - \ell_{01})(1 - \ell_{02})$ and $1 - (1 - \ell_{11})(1 - \ell_{12})$ for priorities 0 and 1 respectively. Therefore the QoS guarantee that can be provided is that priority 0 and priority 1 traffic will face a loss probability of not more than 4.23% and 0.33% respectively. ■

5 Balancing QoS and Power in Multi-hop Wireless Sensor Networks

All the QoS considerations thus far have only implicitly used power or energy. Power issues become crucial especially in sensor networks that deploy battery power for sensing and transmission. In terms of the terminology used thus far, power translates to resource which is directly responsible for node availability in a sensor network. With that understanding, we consider a wireless network which uses battery-powered sensors and transmits using a multi-hop peer-to-peer technology. As an illustration, we show four independent examples of trading off performance and power in various multi-hop wireless sensor domains.

5.1 Gossip-based Information Dissemination

Consider a multi-hop wireless sensor network where the objective of each node is to transmit all the information it has sensed and received to other nodes in a distributed fashion. These transmissions are usually based on what is known as gossip protocols. A gossip protocol typically starts with one node that has sensed information which gets passed along to another node. Now two nodes have that information and they pass it along to two other nodes, and so on. We assume that the communication is line-of-sight or point-to-point as opposed to a broadcast where a node can gossip to all the nodes in its range (this is an immediate extension to the scenario here). We assume that nodes only have local information due to the distributed nature and also that nodes could go down due to several reasons (due to attacks, malfunction or battery running down).

It is crucial to realize that all nodes are continuously sensing and spreading gossip with just the local information. Thus it becomes important to develop a stopping rule to decide when to stop spreading the gossip. If the stopping time is too short, then only a few nodes would get the gossip. This is equivalent to having a poor loss QoS (because not having a gossip is equivalent to loss of information). However, a long stopping time would not only imply wastage of battery power, but it also increases congestion in the network and hence delays. The stopping criterion typically is based on a time-to-live parameter which could be (i) the number of times a gossip has been transmitted by a node, (ii) a threshold number of seconds, or (iii) the number of times gossip has been transmitted to nodes that already posses the gossip (based on acknowledgment packets).

Therefore, given a stopping criterion, analytical models can be built to obtain metrics such as the distribution of the number of nodes that get a gossip, distribution of the time for a certain fraction of the nodes to get the gossip, etc. For example, in Ko and Gautam [13], we consider a highly mobile network wherein the nodes have an equal probability of meeting any of the other nodes during a transmission opportunity. Further, we consider a stopping criteria where a node stops spreading gossip when it encounters another node that already has the gossip. For such a system we show that for any practically sized network (i.e., more that 5 nodes), then irrespective of the number of nodes in the network, the average fraction of nodes that get a gossip is 0.82, a constant. We also compute the average time a gossip lasts (i.e. there is at least one active node spreading it). There are several opportunities to extend the results to other assumptions regarding mobility, transmission and stopping rules.

5.2 Routing in Underwater Sensor Networks

Underwater wireless sensor networks are usually based on hydrophones and geophones that perform point to point communication (and not broadcast). The energy dissipation is based on the distance between the transmitter and the receiver. Although there are several articles on power-aware or energy-aware routing in multi-hope wireless networks, the underwater networks (especially on the ocean bottom) have their own unique characteristics that do not enable the use of existing algorithms. Firstly, the network lifetimes vastly exceed the battery life, so batteries have to be replaced from time-to-time. However, since the batteries are in the bottom of the ocean and not easily accessible, usually a large number of batteries are simultaneously replaced. Thus synchronizing battery failures becomes an important criterion.

In Mohapatra et al [17], we consider a seismic monitoring application of underwater sensor networks. The monitoring is periodic and fairly deterministic. The nodes are laid out on a grid structure and the sink node is in the center of the grid. Since all information finally reaches the sink node, nodes closer to the sink tend to lose battery life faster than others. An interesting observation is that if all nodes send their traffic through the shortest path (which would also result in the minimum consumption of energy), then nodes do not fail at the same time. Instead, if nodes with lesser traffic take a longer path, then the timing of failures can be more-or-less synchronized. Thus the total cost of operation per unit time is minimized. It is also worthwhile to comment that

even in the stochastic sensing and transmission case, since the battery consumption is tiny for each transmission, the battery-life due to a sum of a large number of transmissions is fairly predictable. In summary, this is a rich problem with a radically different set of conditions that its terrestrial counterpart.

5.3 QoS Considerations in Network Coding

The concept of reverse-carpooling using network coding is an efficient way to reduce the number of transmissions in systems where transmissions are the dominant consumer of energy. We assume that the transmission is based on broadcast. As an example, consider three nodes in series. We call them node 1, node R (for relay) and node 2. Nodes 1 and 2 cannot reach each other and hence must transmit only through the relay R. Say node 1 has a binary string x_1 to transmit to node 2 and node 2 has a binary string x_2 to transmit to node 1. First, nodes 1 and 2 transmit x_1 and x_2 to the relay. Then the relay, instead of transmitting x_1 to node 2 and use another transmission to send x_2 to node 1, it codes the strings as $x_1 \oplus x_2$ and transmits once. Since nodes 1 and 2 have $x_1 \oplus x_2$ and the strings they transmitted, they can immediately retrieve x_2 and x_1 respectively.

This is certainly an effective way of reducing the number of transmissions and thereby power consumption. However, in a sensor network information arrive randomly and the relay node may not always have packets from opposite sides to code. Thus a trade-off needs to be made whether it is worth waiting for an opportunity to code or whether it is better to send off a packet uncoded. We are essentially balancing latency against power consumption. In Hsu et al [11] we address this problem and show that there exists a threshold policy so that if the number of packets of one type is fewer than a threshold, we must wait and otherwise we must transmit. With a disclaimer that the problem has certain restrictions in terms of the traffic arrival process and transmissions, we find an interesting result which is that it is not necessary to know how long a packet has been waiting to make that decision. Instead all we need is how many packets are waiting. We are exploring several extensions to this problem of distributed decision making in wireless sensor networks.

5.4 Dynamic Voltage Scaling and QoS

In multi-hop wireless sensor networks with significant processor capabilities, the operating system can be controlled by adjusting the voltage dynamically which results in significant energy savings. This technique is called dynamic voltage scaling (DVS) and simplistically if the voltage setting is high then power consumption is high and vice versa. However, if the voltage setting is high then the processing is also fast, thus the latency is low. For over a decade energy management has been used in mobile and resource-constrained environments that are limited by battery capacities. By appropriately building software interfaces for DVS, one can obtain power savings without compromising on performance (see Flautner et al [8]).

To determine the voltage setting to be used at each node, we can formulate an optimization problem that would minimize the long run average energy cost per unit time subject to satisfying QoS requirements such as average latency. Usually the optimal policy is of threshold-type on the workload. In other words, if there are voltage settings $v_1, v_2, \ldots, v_m$, such that $v_1 < v_2 < \ldots < v_m$, then there exist thresholds on the workload $\theta_1, \theta_2, \ldots, \theta_m$ such that $0 \leq \theta_1 \leq \theta_2 \leq \ldots \leq \theta_m$ so that if the workload at any time is between θ_i and θ_{i+1} then use voltage setting v_i. Thus by using only local information, each of the nodes can appropriately tune their processors and efficiently run the system. Of course opportunities exist for improving these mechanisms by monitoring state information of other nodes, etc.

6 Concluding Remarks

In this chapter abstract models of networks were studied and how to guarantee QoS for them was analyzed. The work focused on methodologies rather than applications. The objective was to develop a set of common tools that are applicable to various types of networks. In particular the study of QoS is important and extends to sensor networks. Although mentioned in the earlier part of the chapter that issues such as pricing and availability are very crucial for customer satisfaction, we have not paid much attention to them in the case studies on determining QoS metrics. However while extending the performance analysis to solve design and control problems, it is very important to take pricing and availability into account as well. Another critical aspect that has been left out of this chapter is degradation and failure of resources. A very active research topic that combines issues of availability, QoS and resource degradation/failure which in total is called survivability or robustness has been given a lot of attention recently by both government and industry. Since degradation and failure cannot be quantitatively modeled very well, survivable or robust system designs end up being extremely redundant. Therefore building cost-effective systems that can be robust or survivable is of utmost importance. Interestingly, sensor networks can be used to address several of the issues.

References

[1] G. Bolch, S. Greiner, H. de Meer and K.S. Trivedi *Queueing Networks and Markov Chains, Modeling and Performance Evaluation with Computer Science Applications.* John Wiley & Sons, 1998.

[2] C. S. Chang and J. A. Thomas. *Effective Bandwidth in High-speed Digital Networks.* IEEE Journal on Selected areas in Communications, 13(6), 1091–1100, 1995.

[3] C. S. Chang and T. Zajic. *Effective Bandwidths of Departure Processes from Queues with Time Varying Capacities.* INFOCOM'95, 1001–1009.

[4] G. de Veciana, C. Courcoubetis and J. Walrand. *Decoupling Bandwidths for Networks: A Decomposition Approach to Resource Management.* INFOCOM'94, 466–473.

[5] A.I. Elwalid and D. Mitra. *Analysis and Design of Rate-Based Congestion Control of High Speed Networks, part I: Stochastic Fluid Models, Access Regulation.* Queueing Systems, Theory and Applications, Vol.9, 29–64, 1991.

[6] A.I. Elwalid and D. Mitra. *Effective Bandwidth of General Markovian Traffic Sources and Admission Control of High-speed Networks.* IEEE/ACM Transactions on Networking, 1(3), 329–343, June 1993.

[7] A.I. Elwalid and D. Mitra. *Analysis, Approximations and Admission Control of a Multi-service Multiplexing System with Priorities.* INFOCOM'95, 463–472, 1995.

[8] K. Flautner, S. Reinhardt, and T. Mudge. Automatic performance setting for dynamic voltage scaling. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 260–271, 2001.

[9] N. Gautam, V. G. Kulkarni, Z. Palmowski and T. Rolski. *Bounds for Fluid Models Driven by Semi-Markov Inputs.* Probability in the Engineering and Informational Sciences, 13, 429–475, 1999.

[10] D. Gross and C. M. Harris (1998). *Fundamentals of Queueing Theory.* 3rd Ed., John Wiley and Sons Inc., NY.

[11] Y. Hsu, S. Ramasamy, N. Abedini, N. Gautam, A. Sprintson, and S. Shakkottai. *Opportunities for Network Coding: To Wait or Not to Wait.* IEEE ISIT, 2011.

[12] G. Kesidis, J. Walrand, and C.S. Chang. *Effective bandwidths for Multiclass Markov Fluids and Other ATM sources.* IEEE/ACM Transactions on Networking, 1(4), 424–428, 1993.

[13] Y.M. Ko and N. Gautam. *Epidemic-Based Information Dissemination in Wireless Mobile Sensor Networks.* IEEE/ACM Transactions on Networking, 18(6), 1738–1751, 2010.

[14] K.R. Krishnan, A.L. Neidhardt and A. Erramilli. *Scaling Analysis in Traffic Management of Self-Similar Processes.* Proc. 15th Intl. Teletraffic Congress, Washington, D.C., 1087–1096, 1997.

[15] V. G. Kulkarni. *Effective Bandwidths for Markov Regenerative Sources.* Queueing Systems, Theory and Applications, 24, 1997.

[16] V. G. Kulkarni, and N. Gautam. *Admission Control of Multi-Class Traffic with Service Priorities in High-Speed Networks.* Queueing Systems, Theory and Applications, 27, 79–97, 1997.

[17] A. Mohapatra, N. Gautam, and R. Gibson. *Combined Routing and Node Replacement in Energy-efficient Underwater Sensor Networks for Seismic Monitoring.* Submitted to IEEE Journal of Ocean Engineering, 2011.