Multiple Access Channel: Combining Information and Queueing Theory

Shashibhushan Borade

Abstract

Multiple access of a single receiver by many transmitters is studied extensively by information theorists and queueing theorists. Information theorists assume that each transmitter has an infinite pool of data be transmitted. They ignore the random nature of messages arrivals. This trivialization enables them to study more accurate models of noise and interference at the receiver. On the other hand, queueing theorists ignore (or trivialize) the noise and interference phenomenon at the receiver. This enables them to isolate the effects of the random arrival of messages. Although multiple access is one of the best understood channels by information theorists, they can not address the issues of delay with their assumptions (since all data is assumed to be already present in the transmitter's data pool). On the other hand, queueing theory might be assuming some things about physical layer, which are either suboptimal or impossible from an information theoretic point of view. This motivates the need of a combined analysis for more fundamental results [3]. In this article, we discuss [1] and [2], which are one of the few prominent results in this largely baron field [4].

1 Introduction

A multiple access channel consists of a set of transmitters, who wish to send information to a common receiver. In both these papers, each new packet to be transmitted is treated as a new user. This means the queues at the transmitters are ignored. This is different from a *real* multiple access channel, where the number of users is fixed with each user transmitting only one packet at a time and may have a queue of packets waiting for transmission. Both papers focus on the Gaussian multiple access channel. In this case, the received signal (in discrete time) is given by

$$y = \sum_{1}^{n} x_k + z \tag{1}$$

where x_k is the transmitted signal of transmitter/user k, the number of users in the system is n and z is a Gaussian noise term with unit variance. Each of the users have power P.

The two papers mainly differ in the following manner. In [1], the system operates in an information theoretically optimal way, i.e. users/packets are be successively decoded giving cleaner channel to the yet undecoded users. The sum capacity of this multiple access system

is $\frac{1}{2}\log(1+nP)$, it is unbounded for n growing to infinity. This causes the average delay to be bounded for any packet arrival rate. In other words, stability is not an issue. The issue of interest then becomes the tradeoff between delay and packet arrival rate. They use a very interesting similarity between job scheduling (which is an operations research and queueing theory area) and information theoretic capacity region of multiple access. This is discussed in the next section.

The second paper treats all other users as noise when decoding any one user. The maximum achievable sum rate for this strategy is, $\frac{n}{2}\log[1+P/(1+P\cdot\overline{n-1})]$. It is because the power from the other $n-1$ users acts as extra noise for every user. Each user now sees a point-to-point Gaussian channel with noise power $1+P\cdot(n-1)$. This assumption is equivalent to separating the receiver into n independent receivers: each trying to decode one of the n users. The sum rate remains bounded for this strategy when n grows to infinity. As a result, this system is stable only if the packet arrival rate is small enough. The delay is unbounded for larger arrival rates. This stability region is found out under the assumption that codewords are *long enough* to ensure error free reception. Nonetheless, if packet lengths are not long enough, we can not assume that each packet is served at a rate $\frac{1}{2}\log[1+P/(1+P\cdot\overline{n-1})]$. To address this issue, results on error exponents are used, which explain the tradeoff between codeword-length and error probability. The maximum possible arrival rate is found out, if an error probability of P_e is tolerable for each packet. In the case where the arrival rate is below this threshold, the tradeoffs between the average delay and P_e requirement are studied. Section 3, discusses this in more detail.

2 Assuming Successive Cancellation of Packets

2.1 Job Scheduling

This approach is based on a particular analogy between job scheduling and multiple access channel. Suppose we start with n jobs to be processed. The i'th job requires τ_i amount of service at time $t=0$. Each job leaves the system when the total service it received equals its initial demand. Without loss of generality, assume that the jobs are numbered such that $\tau_1 \leq \tau_2 \cdots \leq \tau_n$. We have k processors to perform these jobs. Processor j can serve at rate s_j. Assume that the processors are numbered such that $s_1 \geq s_2 \cdots \geq \mathsf{s}_k$. Each processor can serve one job at a time and each job can be served by only one processor at a time. We can preempt the jobs when they are running on a certain processor and allocate them to a different processor.

A policy $\mathcal{P}$ is a rule of allocating the different jobs to the different processors (for example based on their remaining service requirements). Let $\tau_j(t)$ denote the service job j needs after time t. Let C_j denote the completion time of job j: when it leaves the system. The following policy minimizes the average completion time of these n jobs given by $(\sum_{j=1}^{n} C_j)/n$: *At every point of time, shorter jobs should be assigned to the faster processors* [5]. Lets call this STF (*shorter tasks faster*) policy.

For the case of single processor i.e. $k = 1$ this policy is equivalent to the *shortest job first* policy. For case of multiple processors, this is proved by showing that completion time is reduced by assigning shorter job to the faster processor, for any two given jobs and two given processors. This result is used repeatedly to prove the optimality of this policy. This optimal policy has following important aspects.

- As shorter jobs are served faster $\tau_j(t) < \tau_i(t)$ if $\tau_j < \tau_i$ i.e. order of job-lengths is preserved at every instant.
- If $\tau_i < \tau_j$, increasing length of job j does not change the completion time for job i. In other words, longer tasks do not influence the service received by shorter tasks.

Now let each job j require service at a rate S_j instead of having a fixed service requirement as before. The possible service rates are given by this polymatroidal region[1]

$$\text{for all} \quad I \subset \{1, \cdots, n\}, \qquad \sum_{i \in I} S_i \leq \sum_{j=1}^{|I|} s_k \tag{2}$$

System is not stable (queues grow unboundedly) for any rate-vector outside this region. Otherwise, a set of $|I|$ processors will be providing a sum rate more than the sum rate of $|I|$ fastest processors. To prove the achievability, observe that corner points of this region are achievable for some matching between jobs and processors. Other points can be achieved by time sharing between the corner points.

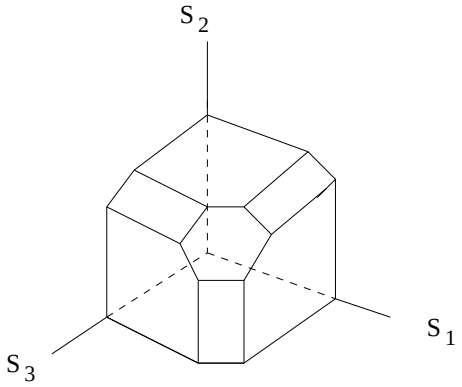

Figure 1: Achievable rate region for $n = 3$.

The above region is exactly same as achievable rate region for a symmetric multiple access channel. For example, Gaussian multiple access with equal power users.

Now consider a job scheduling problem with n jobs of length $\tau_1, \cdots, \tau_n$ and infinite number of processors, each of rate $s_j = \frac{1}{2}\log[1 + P/(1 + P \cdot \overline{j-1})]$ for all $j \in \mathbb{N}$. For this system, let $\sum_{i=1}^{n} C_i$ be the minimum sum completion time under STF policy.

[1]Even when we have less than n processors, this equation can be applied by adding extra 0 rate processors.

In our multiple access system on the other hand, let there be n users/packets where each user j has power P. Each user j has a pool of $\alpha\tau_j$ bits to transmit where α is a scaling constant. These are considered as jobs. We state the following theorem signifying the duality between job scheduling and multiple access.

Theorem 1 *For large enough α, we can send these bits arbitrarily reliably to the receiver such that the sum completion time of these jobs normalized by α is arbitrarily close to $\sum_{i=1}^{n} C_i$.*

Thus the receiver can be considered as having infinite processors, such that processor i is of rate $s_i = \frac{1}{2}\log[1 + P/(1 + P \cdot \overline{i-1})]$ for $i \in \mathbb{N}$. We need large α in the above theorem because large enough code-lengths are required for operating reliably at capacity.

Now consider the case where packets/users are arriving at rate λ (per unit time). Let all the packets be of unit length. This is different from the previous case, where there were a number of packets in the system to start with and no new arrivals after that. The channel is Gaussian multiple access of single sided bandwidth W (instead of being discrete time as before). Let P be the ratio of any transmitter's power to the noise power. The rate of processor j is now modified to $W\log[1 + P/(1 + P \cdot \overline{i-1})]$.

The strategy to minimize the average delay of this system is not known. Lets use the STF strategy anyway. All packets are of unit length, so STF policy is equivalent to *earlier task faster* policy. We can implement the STF strategy with very little feedback from the receiver to transmitters. Each time a new packet enters the system, the receiver conveys the remaining lengths of the packets arrived earlier. Based on this information, the newly arrived packet calculates the departure times for packets arrived earlier[2]. Thus it knows the service rates it is going to receive in future till it departs. It splits its bits accordingly using this information.

For example, the first packet entering the system at $t = 0$ is going to receive rate s_1 till its departure. It will depart at time $C_1 = 1/s_1$. Let the second packet arrive at time $r_1 < C_1$. This packet will split its bits into two parts: first part consisting of $(C_1 - r_1)s_2$ bits and the second part consisting the remaining bits. The first part is transmitted at rate $s_2 = W\log\left(1 + \frac{P}{1+P}\right)$ between time r_1 to C_1. The remaining part is transmitted at rate $s_1 = W\log(1 + P)$. One thing to be noted here is that first packet can remain oblivious to the second packet's arrival as desired. It is because when both packets are being transmitted, the second packet is decoded first treating the first packet as noise. Then second packet's signal is cancelled from the received signal to give the first user an interference-free channel as before.

The normalized average delay versus normalized bit arrival rate for this policy is plotted in the following plot for different power levels (using continuous lines). The bit arrival rate λ is normalized by bandwidth W because doubling both λ and W will not change the delay. Similarly, the average delay is normalized by the bit arrival rate per unit bandwidth so as to give the average delay per bit, instead of per packet.

[2]They are invariant to future packet arrivals due to properties of STF policy earlier discussed.

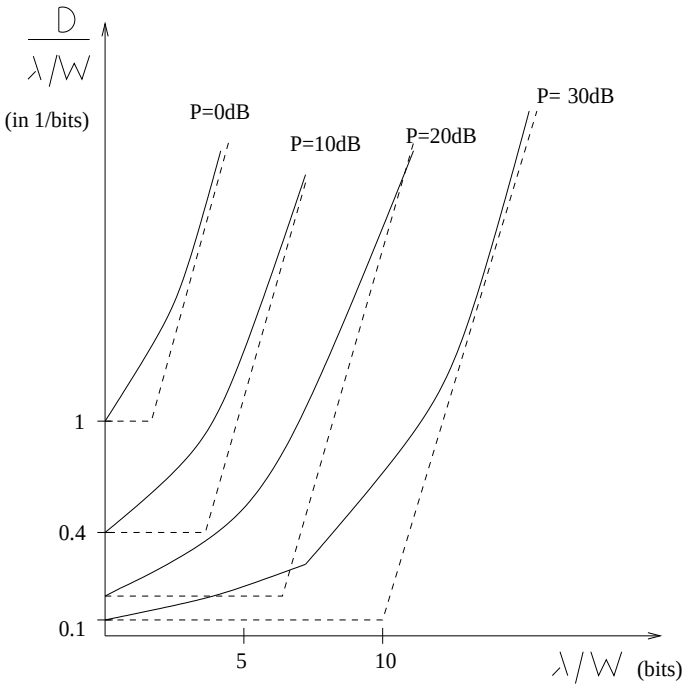

Figure 2: Average delay vs. bit arrival rate

At this point, we do not have any idea how far these curves are from the optimal. Now we get an lower bound on the average delay for arrival rate λ, which is shown by the dashed lines in the above plot. Let p_n be the fraction of time the system has n users. Then the average number of users in the system $\overline{N} = \sum_n np_n$ is related to the average delay $\overline{D}$ by Little's law: $\lambda\overline{D} = \overline{N}$. This is essentially saying that the average rate at which packets are added to the system should equal the rate at which they are emptied from the system for the system to be stable. Observe that when n users are in the system, the total service rate is $\sigma_n = \sum_{i=1}^{n} s_i$. We define continuous function $\sigma(x)$ as the linear interpolation between consecutive points in the set $\{(n, \sigma_n) : \ n \in \mathbb{N}\}$. For stability of the system, the average available service rate $\sum_n p_n\sigma_n$ cannot be less than the average demand rate λ (since all packets have unit length). Using this fact followed by an interesting trick and Jensen's inequality, gives the following lower bound on the average delay as a function of packet arrival rate.

$$\overline{D} \geq \sigma^{-1}(\lambda)/\lambda \tag{3}$$

where σ^{-1} is the inverse function σ. This demonstrates that the STF policy may not be optimal but performs close to optimal at large enough arrival rates.

3 Assuming no Successive Cancellation

Consider a processor-sharing system where the processor's service rate changes with the number of jobs in the system. The processor's rate is given by $\phi(n)$, when there are n jobs sharing the processor. Its rate is equally divided amongst the jobs i.e. each job is served with

rate $\phi(n)/n$. The jobs are arriving in a Poisson process of rate λ. The length of each job is a random variable S with mean $E[S]$. The number of jobs n have a steady state distribution given by,

$$Pr\{n \text{ jobs in the system}\} = \frac{1}{K\phi_!(n)}(\lambda E[S])^n \tag{4}$$

where

$$\phi_!(n) = \prod_{i=1}^{n} \phi(i) \quad \text{and} \quad K = 1 + \sum_{j=1}^{\infty} (\lambda E[S])^j / \phi_!(j)$$

as long as this infinite summation is well defined.

3.1 First order analysis

We think of packets/users as jobs for our multiple access system of bandwidth W. This multiple access system is same as the one considered in previous section. If n users are present, a total communication rate of $nW \ln[1 + P/(1 + P \cdot \overline{n-1})]$ nats/s can be achieved, since every user is decoded by treating other users as noise. In this section, for convenience we have shifted to nats/s instead of bits/s. The length S of each 'job' (i.e. packet) is defined as the packet length in nats divided by bandwidth W. The total rate of this processor (i.e. receiver) when n users are transmitting is defined as $\phi(n) = n \ln[1+P/(1+P\cdot\overline{n-1})]$ nats/s. For convenience, we have divided message length and communication rate by W to define job length S and processor rate $\phi(n)$. Let the average bit arrival rate per unit bandwidth be denoted by $l \equiv \lambda E[S]$.

Note that

$$\lim_{n\to\infty} \phi(n) = \lim_{n\to\infty} n \ln\left(1 + \frac{P}{1 + (n-1)P}\right) = 1$$

Thus the infinite sum $K = 1 + \sum_{j=1}^{\infty} \frac{(\lambda E[S])^j}{\phi_!(j)}$ is well defined if only if $\lambda E[S] = l < 1$. Hence the average number of jobs in the system is finite only if $l < 1$. Using Little's law, we say that $l < 1$ is a sufficient and necessary condition for stability. This tells that maximum throughput of a system of bandwidth W is W nats/s. Average delay of the system will be finite as long as bit arrival rate is smaller than this.

3.2 Analysis for finite packet lengths

The previous analysis assumes that a each packet can be transmitted perfectly at a rate equal to the mutual information between that user and receiver calculated by treating other users as noise. Extremely long codewords might be needed for this. Thus finite length packets can not be transmitted at that rate. To address this issue, they consider the probability of error for a channel whose noise power at time i is σ_i. The noise power may be constant or different over different times i. [6] gives an upper bound on probability of error when

decoding is performed after receiving d symbols of the codeword.

$$P_e \leq \exp\left(\rho \ln M - \sum_{i=1}^{d} E_0(\rho, \sigma_i)\right) \quad 0 \leq \rho \leq 1 \tag{5}$$

here $\ln M$ is the packet length in nats, ρ is any fixed number in $[0, 1]$. $E_0(\rho, \sigma_i)$ is given by,

$$E_0(\rho, \sigma_i) = \rho \ln\left(1 + \frac{P}{(1+\rho)\sigma_i^2}\right)$$

If n_i is the number of users in the system at time i, we have $\sigma_i^2 = 1 + P(n_i - 1)$. Thus the value of $E_0(\rho, \sigma_i)$ only depends on the number of users in the system at time i.

We assume that the receiver always knows the number of users in the system. Lets say that The system needs to achieve a probability of error of P_e. The communication works as follows: each packet in the system starts transmission using an infinitely long Gaussian codeword. The receiver chooses a fixed $\rho \in [0, 1]$ for all time. It counts the RHS of Eq. (5) after each symbol is received. This is possible since the receiver knows the number of users in the system at each time i and thus knows σ_i. As soon as the RHS becomes smaller than P_e, it informs the transmitter to stop transmitting that packet. That packet is now considered as delievered.

This operation gives a new notion of job length obtained by writing Eq. (5) as

$$-\ln P_e + \rho \ln M \geq \sum_{i=1}^{d} E_0(\rho, \sigma_i)$$

The LHS above is the new definition of job length. Note that it depends on the packet's length as well as the P_e requirement. Similarly, the new definition of receiver's total service rate (per unit time) follows from the above equation. If n users are present, it is given by

$$\phi(n) = W\rho n \ln\left(1 + \frac{P}{(1+\rho)(1+(n-1)P)}\right)$$

.

The conditions for stability now follow from Eq. (4). They obtain the best possible tradeoff between probability of error and bit arrival rate. The result obtained in previous subsection follows as a special case of this result. The authors then also find the best average delay for an achievable probability of error requirement.

4 Summary

We have seen two different approaches to combining queueing and information theory. The first approach gives us the asymptotically optimal tradeoff between delay and arrival rate. The packet lengths should be very large for this analysis to hold. A more complete picture of what happens at smaller packet lengths is needed.

The second approach is able to addresses this issue of finite packet lengths very effectively. It gives the tradeoffs between queueing theoretic parameters (arrival rate or delay) and information theoretic parameters (probability of error or bandwidth). Here one should keep in mind that the tradeoffs obtained here are not completely fundamental because a sub-optimal strategy (information theoretically) is used. In spite of being suboptimal, this strategy of treating other users as noise is better than the traditional *collision avoidance* strategy in networking at lower power levels. In any case, it shows one interesting way of bringing queueing theory and information theory together.

One more thing worth mentioning is that queues at the transmitters were ignored by assuming that each packet in the system is a separate user. We do not know the effects of removing this assumption. Incidentally, the different "users" in this multiple access model can be various packets arrived at a single *real* transmitter.

It seems that many of these results can be extended to a Gaussian broadcast system due to its similarity to Gaussian multiple access. It would also be interesting to see what can be said about other network scenarios (other than multiple access) by combining queueing theory and information theory.

References

[1] S. Raj, E. Telatar, D. Tse, "Job scheduling and multiple access," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, to be published.

[2] E. Telatar, R. Gallager, "Combining queueing theory with information theory for multiacess," *IEEE JSAC*, pp. 963–969, Vol. 13, No.6, Aug. 1995.

[3] R. Gallager, "A perspective on multiaccess channels," *IEEE Tran. on Info. Theory*, pp. 124–142, Vol. 31, No. 2, Oct. 1985.

[4] E. Ephremides, B. Hajek, "Information theory and communication networks: an unconsummated union," *IEEE Tran. on Info. Theory*, pp. 2416-2434, Vol. 44, No. 6, Oct. 1998.

[5] T. Gonalez, " Optimal mean finish time preemptive schedules," Technical Report 220, Comp. Sci. Dept., Penn. State University, 1977.

[6] R. Gallager, "Information Theory and Reliable Communication," John Wiley and Sons, New York, 1968.