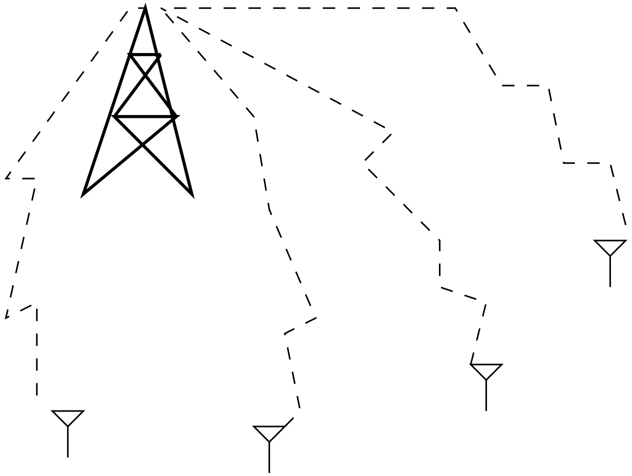
Multiple Access Channel: Combining Queueing and Information Theory

Shashi Borade

Multiple access System



- Uncoordinated transmitters
- Interference: transmitters simultaneously access common receiver
- Zero or little feedback
- Bursty nature of data arrival

Motivation

- For information theory, its a matter of combating noise and interference.

  - More accurate models of noise and interference

  - It ignores the random arrival of data

- For queueing theory, its a matter of

  - Taking advantage of random arrivals , distributed scheduling, resource allocation etc.

  - Trivialized models for noise and interference, which are either suboptimal or impossible

More complete picture is needed!

Tale of two papers: Similarities

- Gaussian multiple access channel

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

  each user has power P and noise power is 1.

- Each packet is treated as an individual user i.e. queues at transmitters are ignored
- Limited feedback available
- Receiver always knows number of users in the system

Tale of two Papers: Differences

- First paper assumes successive cancellation at receiver
  - Information theoretically optimal
  - Sum rate goes to infinity with number of users
  - Stability is not an issue for any packet arrival rate
  - Issue of interest is delay vs. arrival rate tradeoff
- Second paper treats other users as noise when decoding one user
  - Information theoretically suboptimal
  - Sum rate bounded for infinite users
  - Not stable for all arrival rates.

Job Scheduling: Multi Processor Queues

Consider n jobs to be completed. Job j requires τ_j service.

There are k processors. Processor i can serve at rate s_i (per unit time).

At any time instant we can not

- assign any processor to more than one task
- or any task to more than one processor

Although we can

- choose any matching of processors and jobs
- Preempt jobs on one processor and then assign to other processor.

Job j leaves the system when total service s_j is received.

Multi Processor Queues

Their example:

We have n cows

```
          (__)
          (oo)
   /-------\/
  / |     ||
 *  ||----||
    ~~    ~~
```

τ_j = amount of milk in cow j

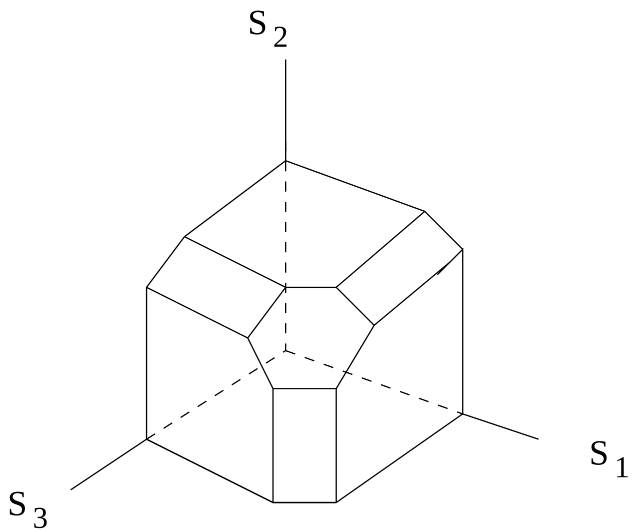
We have k milkmen to milk them

Milkman i can milk s_i per hour.

Achievable rates

Assume $s_1 \geq s_2 \cdots$

If there are only 3 jobs. What possible rates (S_1, S_2, S_3) can be provided?



- Vertices of this polymatroid correspond to some matching between jobs and processors
- Other points achieved by time sharing
- Outside points can not be achieved!

$S_i \leq s_1, \quad S_i + S_j \leq s_1 + s_2, \quad S_1 + S_2 + S_3 \leq s_1 + s_2 + s_3$

Achievable Rates

In general for n jobs $(S_1, S_2 \cdots S_n)$ is achievable iff

$$\text{for every } I \ \subset \ \{1, \cdots n\}, \quad \sum_{i \in I} S_i \leq \sum_{j=1}^{|I|} s_j$$

Note: Add extra 0 rate processors if $k < n$.

Service Policy

Its a rule for assigning processors to jobs.

Say depending on the instantaneous remaining job lengths $\{\tau_j(t)\}$.

Here $\tau_j(t)$ is the remaining length of job j after time t.

Minimizing Average Completion time

Suppose we are given n jobs and we want to minimize the average completion time of the jobs $(C_1 + C_2 \cdots C_n)/n$

Theorem 1 *Assigning shorter tasks to faster processors (at every instant of time) achieves this goal.*

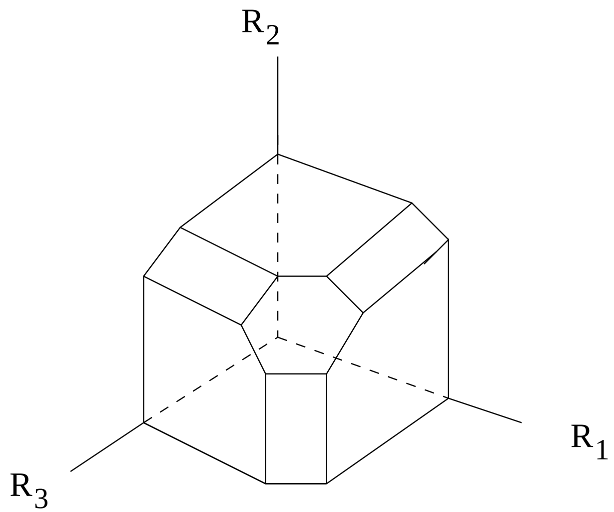
For this policy,

- The order of job lengths is preserved throughout
- Longer tasks do not affect the completion times of shorter tasks.

Gaussian Multiple Access Channel

For Gaussian multiple access channel with users of power P, the capacity region is

$$\text{for every } I \subset \{1, \cdots n\}, \quad \sum_{i \in I} R_i \leq \frac{1}{2}\log(1 + |I|P)$$



Successive cancellation achieves this region.

Signals from already decoded users are subtracted to yield cleaner channel for yet undecoded users.

Gaussian Multiple Access Channel

Let us define

$$\begin{aligned} s_j &= \frac{1}{2}\log(1+jP) - \frac{1}{2}\log(1+(j-1)P) \\ &= \frac{1}{2}\log\left(1+\frac{P}{1+(j-1)P}\right) \end{aligned}$$

Now the capacity region is

$$\text{for every } I \subset \{1,\cdots n\}, \quad \sum_{i\in I} R_i \le \sum_{j=1}^{|I|} s_j$$

Same formula for achievable rate in processor sharing!

So what??

Finite bit pools at transmitters

If each user j has to send a fixed pool τ_j of data to send (no data rate-just fixed number of bits).

Aim is to minimize the average transmission time. Again use *shorter task faster* policy. How?

Finite bit pools at transmitters

If each user j has to send a fixed pool τ_j of data to send (no data rate-just fixed number of bits).

Aim is to minimize the average transmission time. Again use *shorter task faster* policy. How?

User splits its data pool into different parts to be transmitted at different rates.

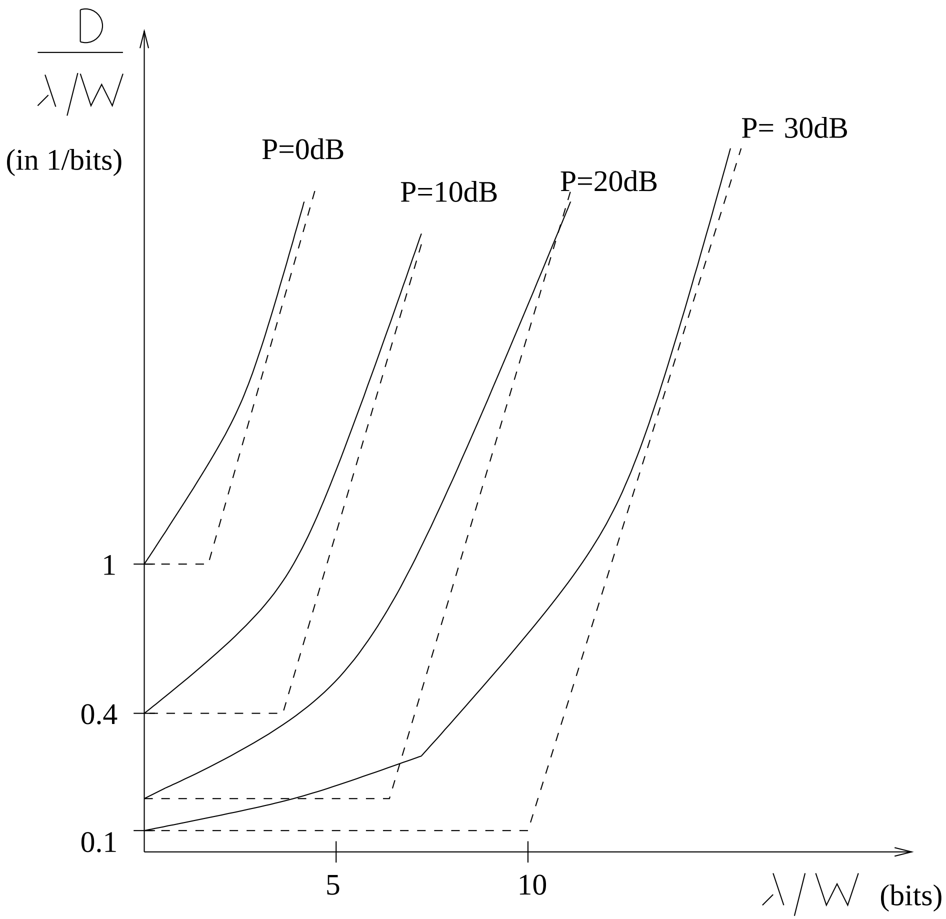
Perform successive Cancellation.

Caution: Each part of a transmitter's data pool should be very long for operating reliably on the capacity region.

Processor sharing with Poisson arrivals

- Now suppose that all jobs/packets are not present as before. Unit length packets are arriving in a Poisson process of rate λ.
- System is stable for all λ as sum rate $\sum_1^n \mathsf{s}_i$ diverges with $n \to \infty$.
- Lets apply *shorter task faster* policy.
  - Its same as earlier task faster.
  - Optimal strategy is not known.
- Each new packet is notified of the remaining lengths of earlier packets.
  - Splits its data accordingly.
  - Possible with very little feedback
- Thanks to successive cancellation, earlier packets can remain oblivious to new arrivals.

Average Delay vs. Arrival Rate



Performance of *shorter task faster* policy

Is this performance any good?

Lower Bound on Average Delay

Let p_n be the fraction of time the system has n users. The average users $\overline{N} = \sum_n p_n n$.

By Little's law average delay $\overline{D} = \overline{N}/\lambda$.

For stability, the long term average service available should be at least the long term average service demand λ.

$$\lambda \leq \sum_n p_n \sigma_n$$

where σ_n is sum service rate for n users i.e. $\sum_{i=1}^{n} s_i$. Let $\sigma(x)$ be the interpolated function between points $(1, \sigma_1), (2, \sigma_2) \cdots$.

Following lower bound on $\overline{D}$ is obtained (see plot).

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

Shorter task faster is close to optimal at high arrival rates.

Part II

Processors Sharing

Consider a processor whose total rate depends on the number of users say $\phi(n)$ for n users. Each user gets $\phi(n)/n$.

Jobs arrivals are rate λ Poisson. Mean length of each job $E[S]$.

The distribution of users in the system,

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

where

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

as long as this infinite summation is well defined.

- Geometric distribution when $\phi(n) = 1$ for all n.

Note Little's law tells that average delay $\overline{D}$ is finite iff expected number of users $\overline{N}$ is finite.

Treating Mutual information as Service rate

- Consider a multiple access channel of bandwidth W. Assume SNR of P (for each user).
- Since all other users are treated as noise, total communication rate of $nW\ln(1+P/(1+P\cdot\overline{n-1}))$ (nats/sec) is achievable. Call this $\phi(n)$.
- Let the packet length in nats be the service demand S of a packet. Thus $\lambda E[S]$ is the average data arrival rate.
- $$\lim_{n\to\infty}\phi(n)=W$$

  For the system to be stable, packets in the system must have a distribution.

This is possible iff $\lambda E[S]/W<1$.

Maximum throughput for system bandwidth W is W nats/s.

Finite Packet lengths

- Transmitting reliably at the rate of mutual information needs very large codewords.
- Small packets can not be sent reliably at that rate.

—

Consider a Gaussian channel where noise power at time i is σ_i^2. The receiver decides to decode the codeword after receiving first d symbols.

The probability of error P_e

$$P_e \leq \exp\left(\rho \ln M - \sum_{i=1}^{d} E_0(\rho, \sigma_i)\right) \quad 0 \leq \rho \leq 1 \tag{2}$$

where $\ln M$ is packet length in nats and ρ is an arbitrary number in $[0, 1]$.

$E_0(\rho, \sigma_i)$ is given by

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

Finite Packet Lengths

As other users are treated as noise, $\sigma_i^2 = 1 + (n_i - 1)P$ where n_i is the number users present at time i.

The network operation is as follows:

- Required P_e is specified for the system. Receiver chooses any fixed $\rho \in [0, 1]$ for all time.
- Each packet starts sending infinite length codeword.
- After each symbol, the receiver calculates the RHS of Eq. (2). As soon as it is smaller than P_e, informs the transmitter to stop transmitting that codeword
  - Limited feedback needed.

Finite Packet Lengths

This network operation gives a new notion of job lengths obtained by rewriting Eq. (1) as

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

LHS above is the defined as demand S of packets.

- It depends on both packet length and P_e requirement.

The service rate (per unit time) is defined as

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

With these definitions of demand and service rate, the best possible tradeoff between P_e and bit arrival rate $\lambda E[\ln M]$ follows from Eq. (1).

– Previous result follows as a special case of this.

Concluding Remarks

- First approach gives asymptotically optimal tradeoff between delay vs. arrival rate

  - Packet lengths need to be very large.

- - Second approach, gives tradeoffs between queueing theory parameters (delay, arrival rate) and information theoretic parameters (bandwidth, probability of error, SNR)

  - Addresses the finite packet length issue effectively

  - Nonetheless, it is based on a suboptimal strategy.

- What if transmitter queues are not ignored?

- Some easy extensions to Gaussian broadcast channels.