ECE 5325/6325: Wireless Communication Systems Lecture Notes, Spring 2013

Lecture 3

Today: (2) Trunking

- Reading: Today: 4.2.2. Thu: Rap 3.7.2 (pdf on Canvas).

1 Trunking

Trunking refers to sharing few channels among many users. Let N_C be the number of channels available in a cell (we called it K last lecture, my bad), and n_{user} be the number of channels per cell. Each user requires a channel infrequently, so a dedicated channel for each user is not required. But, the request for a channel happens at random times, so for any N_C less than the number of users, it is possible that there will be more requests than channels.

- *Average holding time*: Average call duration, denoted H.
- *Call rate*: Average number of calls a user makes per unit time, denoted λ. Typically taken to be at the busiest time of day.
- *User traffic intensity*: The average user requests this much traffic, $A_u = \lambda H$. This is really unitless (Hz times sec) but is said to have units of "Erlangs", where 1 Erlang is the traffic to fill one channel all of the time.
- *Total offered traffic intensity*: The total amount of traffic users in the cell request of the system, denoted T_{tr}, equal to the number of users per cell times A_u.
- *Probability of blocking* $\Pr_{block}$ or "Grade of Service" (GOS): The probability an offered call will be blocked (and thus not served, or carried by the system).
- *Number of channels per cell* called N_C in Molisch.

For example, if a user makes on average, two calls per hour, and that call lasts an average of 3 minutes, $A_u = \frac{2}{60 \text{ min}} 3 \text{ min} = 0.1$ Erlang. (Check your units!)

Then, to compute the total offered traffic intensity, and the total offered traffic intensity per channel (denoted A_c),

$$T_{tr} = n_{user} A_u, \quad A_c = A/N_C$$

For the above example, assume that there are 1000 users and 200 channels. Then $T_{tr} = 1000(0.1) = 100$, and $A_c = 100/200 = 0.5$.

Note that A_c is a measure of the *efficiency* of the utilization of the channels.

How should we design our system? Obviously, A_c should be less than one ($T_{tr} < N_C$); or we'll never satisfy our call demand. But how should we set n_{user}, A_u, N_C to satisfy our customers?

First choice: what do we do when a call is offered (requested) but all channels are full?

- *Blocked calls cleared*: Ignore it.
- *Blocked calls delayed*: Postpone it!

1.1 Blocked calls cleared

1. Call requests are a Poisson process. That is, the times between calls are exponentially distributed, and independent from each other.
2. Call durations are also exponentially distributed.
3. If a user is blocked and cleared, she will not immediately call back, instead, her new call will be at a statistically independent time (she gives up on the call).

These assumptions lead to the **Erlang B formula**:

$$\Pr_{block} = \frac{T_{tr}^{N_C}/C!}{\sum_{k=0}^{N_C} T_{tr}^k/k!} \tag{1}$$

Since N_C may be very high, it is typically easier to use a chart to determine $\Pr_{block}$. Figure 17.2 in Molisch is one example. I am providing Figure 3.6 from [1] which has a few more data points. By setting the desired $\Pr_{block}$, we can derive what number of channels we need; or the maximum number of users we can support (remember $T_{tr} = n_{user}A_u$); or the maximum A_u we can support (and set the number of minutes on our calling plans accordingly).

1.2 Blocked calls delayed

Instead of clearing a call; put it in a queue (a first-in, first-out line). Have it wait its turn for a channel. ("Calls will be processed in the order they were received"). There are now two things to determine

1. The probability a call will be delayed (enter the queue), and
2. The probability that the delay will be longer than t seconds.

The first is no longer the same as in (1); it goes up, because blocked calls aren't cleared, they "stick around" and wait for the first open channel.

Here, we clarify the meaning of "grade of service" for a blocked calls delayed system. Here it means the probability that a call will be forced into the queue AND it will wait longer than t seconds before being served (for some given t).

We need a couple additional assumptions:

1. The queue is infinitely long. In a computer system, this translates to infinite memory.
2. No one who is queued gives up / hangs up (rather than wait).

With these assumptions, we can derive the Erlang C formula, for the probability that a call will be delayed:

$$P\left[\text{delay} > 0\right] = \frac{T_{tr}^{N_C}}{T_{tr}^{N_C} + N_C!\left(1 - T_{tr}/N_C\right)\sum_{k=0}^{N_C-1} T_{tr}^{k}/k!} \tag{2}$$

It is typically easiest to find a result from a figure. I am providing Figure 3.7 from [1]. Once it enters the queue, the probability that the delay is greater than t (for $t > 0$) is given as

$$P\left[\text{delay} > t|\text{delay} > 0\right] = \exp\left(-\frac{N_C - T_{tr}}{H}t\right) \tag{3}$$

The two combined are needed to find the marginal (overall) probability that a call will be delayed AND experience a delay greater than t, the event that we are quantifying in $\text{Pr}_{delay>t}$.

$$\begin{aligned}\text{Pr}_{delay>t} &= P\left[\text{delay} > t|\text{delay} > 0\right] P\left[\text{delay} > 0\right] \\ &= P\left[\text{delay} > 0\right] \exp\left(-\frac{N_C - T_{tr}}{H}t\right) \end{aligned} \tag{4}$$

Example: $N = 7$ cell cluster
A 7 cell cluster (with $N = 7$) has 30 MHz allocated to it for forward channels and each channel is 200 kHz. Assume blocked-called-delayed and a probability of delay of 1%, and each user makes one 10 minute call every 3 hours. (a) What is the number of users that can be supported? (b) What is $\text{Pr}_{delay>10}$ seconds? (c) What if it was a blocked-calls-cleared system with Pr_{block} of 1%?

1.3 Discussion

What are the problems or benefits we see from the assumptions we've made? Are call requests "memoryless"? Is the exponential interarrival time assumption accurate? When catastrophic events occur, or major news breaks, what happens? How should a communications system be designed to handle these cases?

References

[1] T. Rappaport. *Wireless Communications: Principles and Practice.* Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

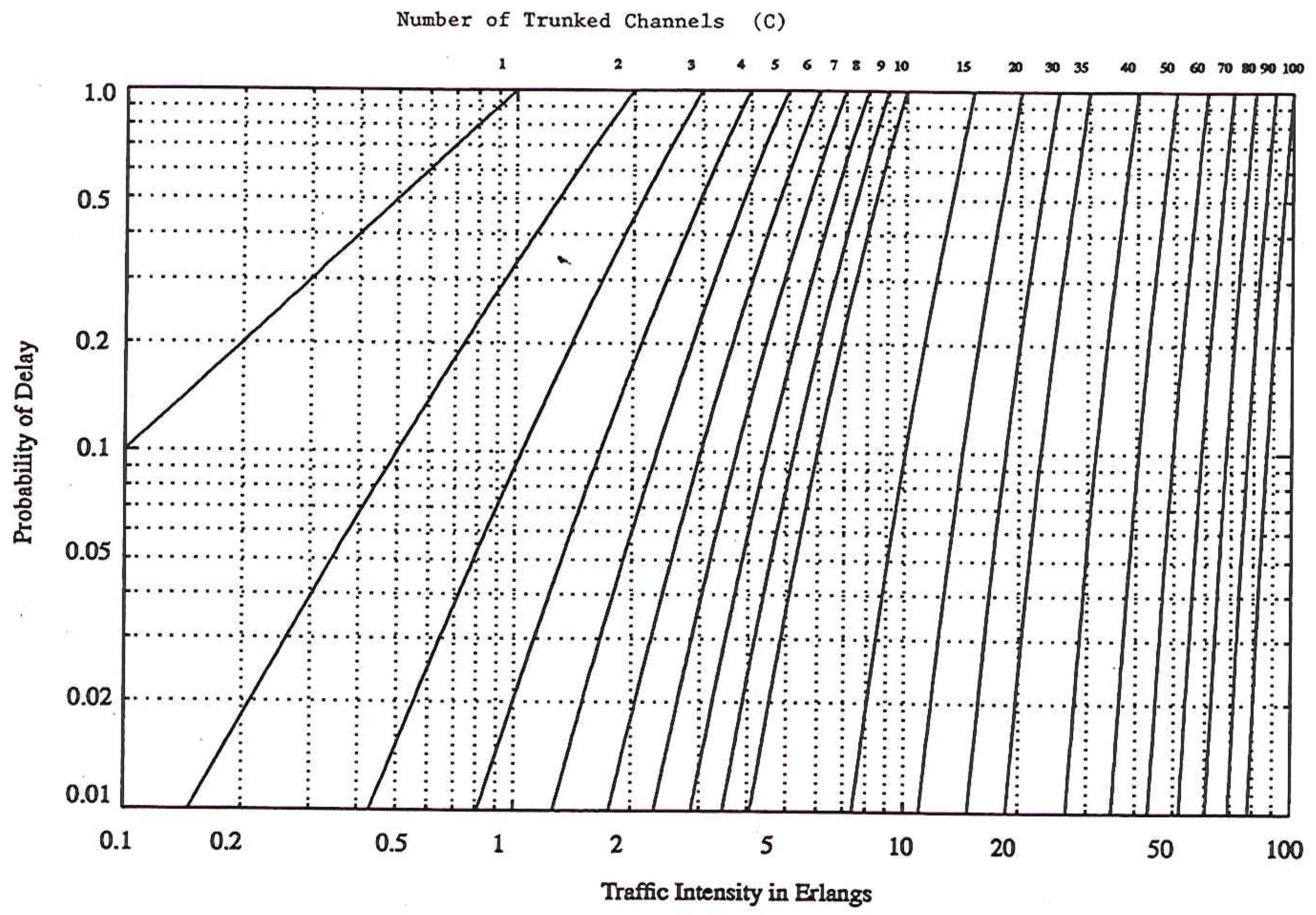

Figure 3.7 The Erlang C chart showing the probability of a call being delayed as a function of the number of channels and traffic intensity in Erlangs.

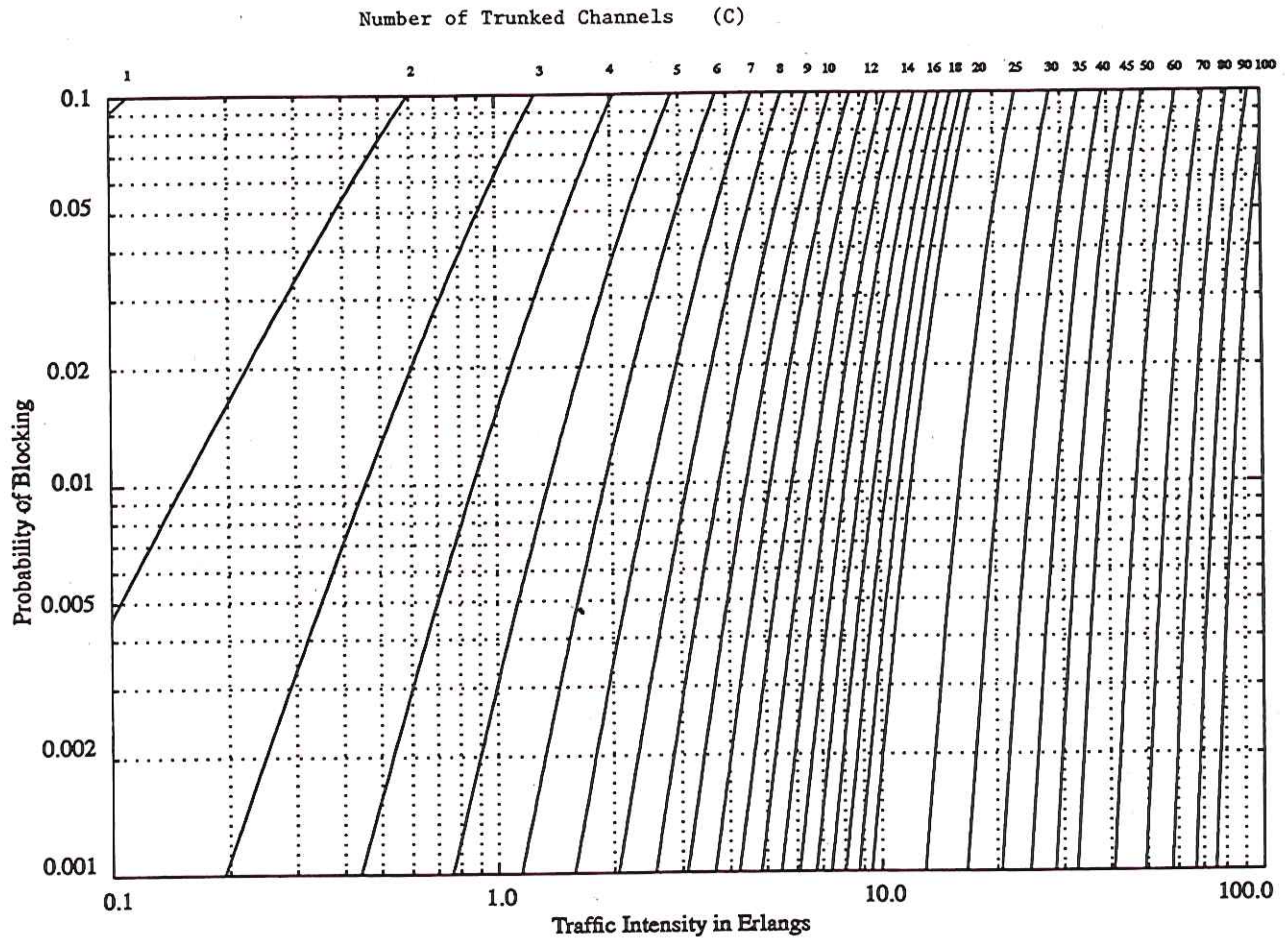

Figure 3.6 The Erlang B chart showing the probability of blocking as functions of the number of channels and traffic intensity in Erlangs.