ERLANG-B FORMULA & TABLE

A Tool for Traffic, Capacity, and Blocking Probability

EEEN 567 – SATELLITE ENGINEERING

Wednesday, January 14, 2026

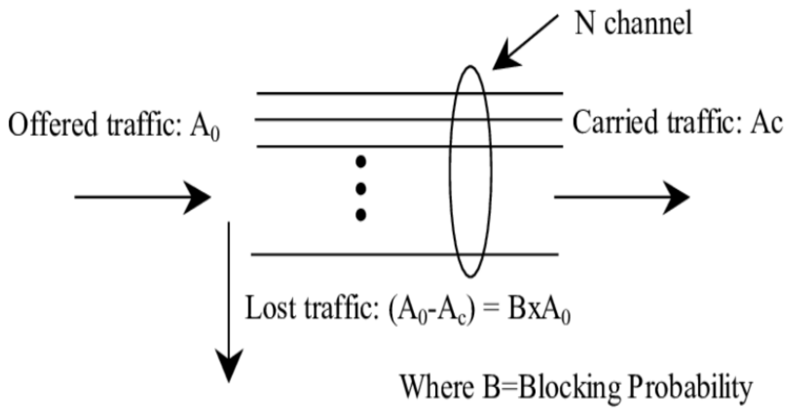
WHY DO WE NEED ERLANG B FORMULA?

1. Core Question:

- How many channels do we need to serve many users, when not all users call at once?

2. The Challenge (The Blocking Scenario):

- You are designing a satellite communication system.
- You have 1000 potential users, but they make calls randomly.
- If all channels are busy when a new call arrives, that call is **blocked** (gets a "**busy signal-Mteja hapatikani**").
- Adding infinite channels is wasteful and expensive.



3. Solution:

- Find the optimal number of channels to keep blocking acceptably low (e.g., 1% or 2%).

4. This is the problem the Erlang B formula solves.

DEFINITIONS – TRAFFIC INTENSITY

1. **Traffic Intensity (A)** is the average number of concurrent calls over a period.

2. **Formula:**

$$A = \lambda * h$$

where

λ = Call arrival rate (calls per hour)

h = Average call holding time (hours per call)

3. **Example:** 30 calls/hour, each lasting 3 minutes (0.05 hours).

$$A = 30 * 0.05 = 1.5 \text{ Erlangs}$$

DEFINITIONS – GRADE OF SERVICE (GoS)

1. **Grade of Service (GoS)** is the probability of blocking (Pb) a new call due to congestion.
   - A target performance metric (e.g., P_b = 0.01 or 1%).
2. **Capacity (N) is t**he number of available servers/channels/circuits.

ERLANG B FORMULA /01

The Mathematical Model:
Erlang B formula calculates the probability that all N channels are busy, assuming:

1. **Pure Loss System:** Blocked calls are cleared (user hangs up , i.e no waiting calls).
2. **Infinite Sources:** Many independent users.
3. **Random Arrivals:** Calls follow a Poisson process.
4. **Exponential Holding Times.**

ERLANG B FORMULA /01

1. The Formula:

$$P_b(N, A) = \frac{A^N / N!}{\sum_{k=0}^{N} A^k / k!}$$

where:

P_b = Probability of Blocking (Grade of Service)

A = Offered Traffic (Erlangs)

N = Number of Channels

2. Problem: This is tedious to calculate by hand for every design!

ENTER THE ERLANG B TABLE

The Engineer's Solution:
A pre-calculated table that gives the relationship between **ANY TWO** of the three variables:

1. **Offered Traffic (A)** in Erlangs
2. **Number of Channels/Servers (N)**
3. **Grade of Service (Blocking Probability, P_B)**

HOW TO READ ERLANG B TABLE

1. **Choose your desired GoS** (e.g., 0.01, 0.02) – this is your *column*.
2. **Know your traffic (A)** – find it in the body of the table to get required **N**.
3. **Or, know your channels (N)** – find the corresponding **A** you can support.

ANATOMY OF ERLANG B TABLE

N	$P_b = 0.01$	$P_b = 0.02$	$P_b = 0.05$
1	0.0101	0.0204	0.0526
5	1.361	1.657	2.218
10	4.461	5.092	6.216
15	8.108	9.010	10.63
20	12.03	13.19	15.25

INTERPRETATION:

1. For N=10 channels and a strict GoS of 1% (0.01): You can support 4.461 Erlangs of traffic.
2. If you have 5 Erlangs of traffic and need GoS of 2%: You need at least N=10 channels (5.092 Erlangs capacity > 5).

WORKED EXAMPLE

Design the voice channels for a satellite telephone system with following traffic parameters:

- Forecast Traffic Intensity, λ = 60 calls/hour.
- Average call duration h = 3 minutes.
- Target Grade of Service: 1% blocking .

SOLUTION

Given:

- Traffic intensity, λ = 60 calls/hour
- Average call duration h = 3 minutes (0.05 hrs).
  Target Grade of Service: 1% blocking (P_b = 0.01).

Step 1: Calculate Offered Traffic (A)

A = λ * h = 60 calls/hr * 0.05 hr/call = 3 Erlangs

Step 2: Use Erlang B Table (P_b = 0.01 column)

Find the smallest traffic entry ≥ **3 Erlangs**.

- N=5 → Supports 1.361E (Not enough)
- N=8 → Supports 3.128E (Enough!)
- N=7 → Supports 2.501E (Not enough)

Conclusion:

You need **N = 8 traffic channels** to handle 3 Erlangs at 1% blocking.

APPLICATIONS IN ELECTRICAL ENGINEERING

1. **Traditional Telephony:** Trunk (T1/E1 line) dimensioning between switches.
2. **Cellular Systems (2G/3G/4G):** Sizing voice traffic channels in a sector.
3. **Satellite Communications:** Calculating required transponders for customer traffic.
4. **Call Centers:** Determining the number of agents/phone lines needed.
5. **Critical Infrastructure:** Designing reliable dispatch systems (police, fire).

Core Engineering Principle: It enables the **quantitative trade-off** between **Quality of Service (QoS)** and **Capital Cost** (infrastructure).

LIMITATIONS & MODERN CONTEXT

1. **Limitations of the Basic Model:**
   a) **Assumes blocked calls disappear.** (Erlang C models queuing, like in call centers).
   b) **Does not account for data traffic** (bursty, variable rate).
   c) **Modern 4G/5G data networks use packet-switched models** (e.g., based on bandwidth and delay).
2. **But It's Still Foundational!**
   a) **Voice-over-LTE (VoLTE)** still requires circuit-switched-like resource reservation.
   b) **Satellite and legacy system design** relies heavily on it.
   c) The **core concept of stochastic traffic engineering** is universal.
   d) Modern tools (software, scripts) still implement the Erlang B formula at their core.

SUMMARY & KEY TAKEAWAYS

1. **Purpose:** Erlang B is a **loss probability model** for dimensioning circuit-switched networks.
2. **Key Variables: A** (Traffic in Erlangs), **N** (Channels), $\mathbf{P_b}$ (Blocking Probability).
3. **The Table:** A vital engineering tool to quickly solve $P_b = f(A, N)$ for any two variables.
4. **Design Workflow:**
   a. Forecast traffic (λ, h) → Calculate **A**.
   b. Choose target $\mathbf{P_b}$ (GoS).
   c. Use Table to find required **N**.
5. **Bigger Picture:** It embodies the fundamental engineering compromise between **system cost** and **service quality**.