Home Work Report:

Modeling and Analysis of Multi-Service Erlang Loss Systems

Name: *Dana Dagher*

Course: *B22 – Queuing Theory*

Instructor: *Prof. Salah Din El Ayoubi*

GitHub link: https://github.com/Dana-Dagher/multiservice-erlang-project

1. Introduction

This project studies resource dimensioning in a circuit-switched network serving heterogeneous traffic. We quantify blocking probabilities and admissible arrival rates under a strict capacity constraint, first for voice-only traffic and then for a mixed voice+video traffic mix. Starting with the voice-only case clarifies baseline capacity and validates numerical methods before their application to the multi-class problem, ensuring the extension is built on a correct foundation.

2. System model and parameters

Parameters (used throughout):

- Total circuits : C=40
- Voice: c_v =1 circuit, mean T_v =180 s → μ_v =1/180 s^{-1}
- Video: c_s =5 circuits, mean T_s =120 s → μ_s =1/120 s^{-1}
- Arrival rates: voice $\lambda_{v(variable)}$,video λ_s=0.2λ_v

State variables: n_v = number of active voice calls; n_s = number of active video calls.
Capacity constraint: $c_s n_s + c_v n_v \leq C$

Blocking definition: A new class-iii call is blocked in state (n_s,n_v) if adding c_i circuits violates the capacity constraint.

- Voice blocked if $c_s n_s + c_v n_v + c_v > C$.
- Video blocked if $c_s n_s + c_v n_v + c_s > C$.

3. Methods and implementation (order emphasized)

Important: the analysis explicitly **starts with voice-only** (Sections Q1 and Q2) to obtain a validated baseline and then **extends to voice+video** (Sections Q3 and Q4). This progression ensures the multi-class model inherits a numerically validated single-class implementation.

Q1 — Erlang-B (voice-only)

For a single-class loss system the blocking probability is $B = \frac{\frac{A^N}{N!}}{\sum_{i=0}^{N} \frac{A^i}{i!}}$ where A= λ_v/μ_v

Implementation notes:

- Sweep λv on a grid (0.01–0.5 calls/s), compute offered traffic AAA and B(C,A) .
- Find λv such that B=0.01 via interpolation.
- Output: baseline λ_v for 1% blocking.

Q2 — CTMC verification (voice-only)

Model: birth–death CTMC with states n=0,…,C

- Arrival n→n+1 rate λ_v (if n<C).
- Departure n→n−1 rate $n\mu_v$ (if n>0).

Implementation notes:

- Build generator matrix Q(size C+1 by C+1), diagonals set to negative row sums.
- Solve steady-state by solving $\pi Q=0$ with normalization $\sum \pi i=1$. Practical approach: solve $A\pi^T= b$ with $A= Q^T$ and last row replaced by ones.
- Blocking = π_C . This verifies the Erlang-B baseline numerically.

Q3 — CTMC enumeration and two-class CTMC (voice + video)

After validating voice-only results, extend to mixed traffic.

State enumeration:

>Enumerate all (ns,nv) and include only states satisfying $c_s n_s+c_v n_v \le C$. Denote the total number of states by N.

Generator matrix construction (sparse):

- For each state k corresponding to (ns,nv) :
    - If $c_s n_s+c_v n_v +c_v \le C$, add transition to (ns,nv+1) at rate λv ;else mark state as voice-blocking.
    - If $c_s n_s+c_v n_v +c_s \le C$, add transition to (ns+1,nv) at rate λs ;else mark state as video-blocking.
    - If nv>0, add departure to (ns,nv−1) at rate nvμv .
    - If ns>0 , add departure to (ns−1,nv) at rate nsμs .
    - Set diagonal $Q(k,k)=-\sum_{j\neq k} Q(k,j)$.

Steady-state:

- As before, solve $\pi Q=0$ with normalization. We used sparse storage for Q to reduce memory and speed up linear solves.

Blocking probabilities:

- Bv(λv) = sum of π over states marked voice-blocking.
- Bs(λv) = sum of π over states marked video-blocking.

Q4 — Analytical multi-Erlang (product-form) validation

Analytical formula for the joint steady-state on feasible states A:

$$p(n_s,n_v)=\frac{1}{G}E_s^{n_s}/_{n_s!}\;E_v^{n_v}/_{n_v!}$$

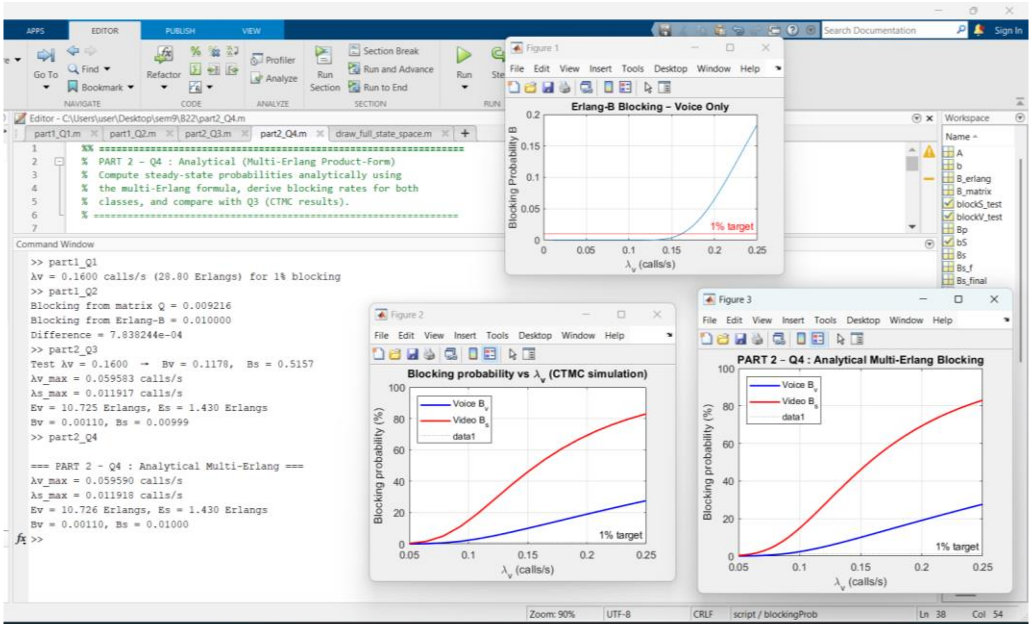
onstant:

$$G=\sum_{(n_s,n_v)\in A} E_s^{n_s}/_{n_s!}\;E_v^{n_v}/_{n_v!}$$

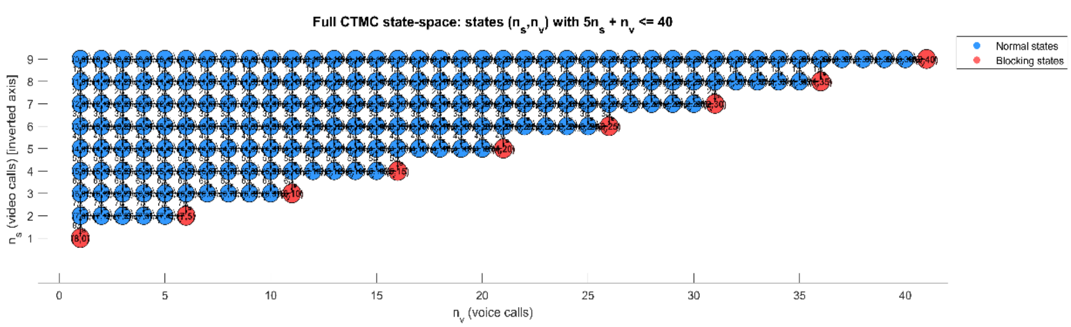
Blocking probabilities defined as in Q3 and computed by summation of π(ns,nv) over the appropriate blocking states. Use the same bisection method to compute λv that satisfies both blocking constraints. This provides a fast analytical validation of the CTMC results.

4. Results (numerical outputs)

All reported values are produced by the MATLAB scripts in the repository.



The `draw_full_state_space.m` script enumerates all feasible states (ns,nv) with 5ns+nv≤40 , draws them on a grid, and highlights blocking states (where capacity is full) in red.
Arrows annotate feasible arrival and departure transitions (voice/video arrivals and multi-rate departures);



5. Discussion and interpretation

- **Start-with-voice approach:** Beginning with voice-only analysis was essential to validate numerical CTMC implementation against the closed-form Erlang-B benchmark. The validated CTMC code was then safely extended to the two-class problem.
- **Impact of video traffic:** Introducing video streams (each consuming 5 circuits) reduces admissible voice arrival rate dramatically: λv drops from ~0.16 to ~0.0596 calls/s — a reduction of roughly 63%. This illustrates how multi-rate demands drastically affect capacity planning.
- **Blocking balance:** At the computed operating point voice blocking is ≈ 0.11% and video blocking ≈ 0.99%, both meeting the 1% QoS objective. Video blocking is the tighter constraint due to its larger circuit demand.
- **Method comparison:** The CTMC provides explicit state-space insight and visualizable structure, while the analytical multi-Erlang product-form is computationally efficient for many parameter sweeps. The near-perfect agreement validates both approaches.
- **Numerical notes:** small differences arise from linear system solves (floating point rounding), bisection tolerances, and factorials in product-form computations.