```
(* Modelo matemática, obtido através da utilização de Cadeias de Markov,
para a caracterização da probabilidade individual de transmissão,
débito e tempo de serviço do protocolo MAC proposto para uma
  Rede Cognitiva descentralizada "multi-channel" - MC-C2RMAC. *)


ClearAll["Global`*"]

solSU = Solve[
    IDLE == IDLE * (1 - Pp * PChFree) + COMP * (1 - PChAss * PCts + PChAss * PColl * PCts) *
         (1 - PChFree) + TRANS * PEnd * (1 - Pp) &&
     COMP == COMP * (1 - PChAss * PCts + PChAss * PColl * PCts) * PChFree +
        IDLE * Pp * PChFree + TRANS * PEnd * Pp &&
     TRANS == TRANS * (1 - PEnd) + COMP * PChAss * (1 - PColl) * PCts &&
     IDLE + COMP + TRANS == 1, {IDLE, COMP, TRANS}] // FullSimplify
```

```
(* Automatizado para qualquer que seja o numero de canais*)
n = 10;
mu = 5;
cw = 10;
l = 3;
PCts = 1;
Pp = 1;
PI = {0.5, 0.6, 0.7};

MeanPoisson = Sum[1 / (mu / PI[[k]] + 1), {k, 1, l}];
PChFree = 1 - PDF[PoissonDistribution[MeanPoisson], 0];

nSendRts =
  N[Sum[PDF[PoissonDistribution[MeanPoisson], k] * (n - (l - k)), {k, 1, l}]] /
   PChFree;

PColl = 1 - cw * (1 / cw) * (1 - (1 / cw)) ^ (nSendRts - 1);

PChAss =
  If[(Sum[PDF[PoissonDistribution[MeanPoisson], k] * (k), {k, 1, l}] / PChFree) /
      (nSendRts * (1 - PColl)) ≥ 1, 1, Sum[PDF[PoissonDistribution[MeanPoisson], k] *
       (k) / (nSendRts * (1 - PColl)), {k, 1, l}] / PChFree];

PEnd = 1 / ((1 / l) * Sum[mu / PI[[k]], {k, 1, l}]);

MeanServiceTime = 1 / ((COMP /. solSU[[1]]) * PChAss * (1 - PColl));

Throughput = Sum[mu / ((mu / PI[[k]]) + 1), {k, 1, l}];

Print["\nThroughput:       ", N[Throughput]];
Print["MeanServiceTime: ", N[MeanServiceTime]];
Print["SendRts:          ", N[nSendRts]];
Print["PChFree:          ", N[PChFree]];
Print["PSucc:            ", N[1 - PColl]];
Print["PChAss:           ", N[PChAss]];
Print["\nIDLE: ", N[IDLE /. solSU[[1]]], " COMP: ",
  N[COMP /. solSU[[1]]], " TRANS: ", N[TRANS /. solSU[[1]]]];
Print["\nIDLE  -> IDLE:      ", N[1 - PChFree]];
Print["IDLE  -> COMP:      ", N[PChFree]];
Print["COMP  -> IDLE:      ", N[(1 - PChAss + PChAss * PColl) * (1 - PChFree)]];
Print["COMP  -> COMP:      ", N[(1 - PChAss + PChAss * PColl) * PChFree]];
Print["COMP  -> TRANS:     ", N[PChAss * (1 - PColl)]];
Print["TRANS  -> COMP:     ", N[PEnd]];
Print["TRANS  -> TRANS:    ", N[1 - PEnd]];




ClearAll["Global`*"]

solSU = Solve[IDLE == IDLE * (1 - Pp * PChFree) +
        COMP * (1 - PChAss + PChAss * PColl) * (1 - PChFree) + TRANS * PEnd * (1 - Pp) &&
      COMP == COMP * (1 - PChAss + PChAss * PColl) * PChFree + IDLE * Pp * PChFree +
        TRANS * PEnd * Pp && TRANS == TRANS * (1 - PEnd) + COMP * PChAss * (1 - PColl) &&
      IDLE + COMP + TRANS == 1, {IDLE, COMP, TRANS}] // FullSimplify;

mu = 5;
Pp = 1;
```

```
l = 3;
PCts = 1;
PI = {0.5, 0.6, 0.7};

cwTeo = {};
txTeo = {};
MeanServiceTimeTeo = {};

For[n = 3, n < 50, n = n + 2;
 For[cw = 0, cw < 20, cw = cw + 2;

  Clear[MeanPoisson, PChFree,
   nSendRts, PSucc, PColl, PChAss, PEnd, MeanServiceTime];

  MeanPoisson = Sum[1 / (mu / PI[[k]] + 1), {k, 1, l}];
  PChFree = 1 - PDF[PoissonDistribution[MeanPoisson], 0];

  nSendRts =
   N[Sum[PDF[PoissonDistribution[MeanPoisson], k] * (n - (l - k)), {k, 1, l}]] /
    PChFree;

  PSucc = cw * (1 / cw) * (1 - (1 / cw)) ^ (nSendRts - 1);
  PColl = 1 - PSucc;

  PChAss = Sum[PDF[PoissonDistribution[MeanPoisson], k] *
       (k) / (nSendRts * PSucc), {k, 1, l}] / PChFree;

  PEnd = 1 / ((1 / l) * Sum[mu / PI[[k]], {k, 1, l}]);

  MeanServiceTime = 1 / ((COMP /. solSU[[1]]) * PChAss * (1 - PColl));

  If[cw == 2, cwTeo = {cw}, cwTeo = Append[cwTeo, cw]];
  If[cw == 2, txTeo = {TRANS /. solSU[[1]]},
   txTeo = Append[txTeo, TRANS /. solSU[[1]]]];
  If[cw == 2, MeanServiceTimeTeo = {MeanServiceTime},
   MeanServiceTimeTeo = Append[MeanServiceTimeTeo, MeanServiceTime]];

  Export[
   "/Users/glxitu/Dropbox/Doutoramento/Artigos/MChannel/validation/exprnd/ch"
    <> ToString[l] <> "/n" <> ToString[n] <> ".m",
   Transpose[{cwTeo, txTeo, MeanServiceTimeTeo}], "Table"];

 ];
]
```