```
(* 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 "single-channel" - C2RMAC. *)


ClearAll["Global`*"]

(*SU e AP sem valores*)

solSU = Solve[IDLESU == IDLESU * (1 - Palpha) + CW1 * (1 - P2st * (Piap * (Pii - 1) + 1)) +
        CW11 * (Piap * (1 - Pii)) + CW2 * (1 - Pbeta) + TXSU * (1 - Pii * Pgamma) &&
      CW1 == IDLESU * (Palpha) + TXSU * (Pii * Pgamma) &&
      CW11 == CW1 * (P2st * (1 - Piap)) + CW11 * (1 - Piap) &&
      CW2 == CW1 * (Pii * Piap * P2st) + CW11 * (Piap * Pii) &&
      TXSU == CW2 * Pbeta &&
      IDLESU + CW1 + CW11 + CW2 + TXSU == 1,
     {IDLESU, CW1, CW11, CW2, TXSU}] // FullSimplify;

Refine[solSU /. (Piap + P2st * Palpha * Pp +
   Piap * ((-P2st) * Pbeta * Pgamma * Pii^2 +
    Palpha * (1 + P2st * (-1 + Pii + Pbeta * Pii))) * Pp) → a]

solAP = Solve[IDLEAP == IDLEAP * (1 - Piap) + Pt1 * (1 - Psu1) * (1 - Piap) +
        Pt2 * (1 - Psu2) * (1 - Piap) + Pt2 * Psu2 * (1 - Piap) && Pt1 ==
       IDLEAP * Piap + Pt1 * Piap * (1 - Psu1) + Pt2 * Piap * (1 - Psu2) + Pt2 * Psu2 * Piap &&
      Pt11 == Pt1 * Psu1 * (1 - Piap) + Pt11 * (1 - Piap) &&
      Pt2 == Pt1 * Piap * Psu1 + Pt11 * Piap &&
      IDLEAP + Pt1 + Pt11 + Pt2 + ((cw2bAvg * Pt2 * Psu2) / Piap) == 1,
     {IDLEAP, Pt1, Pt11, Pt2}] // FullSimplify;

Refine[solAP /. (1 + Psu1 + cw2bAvg * Psu1 * Psu2) → b]


ClearAll["Global`*"]

solSU = Solve[IDLESU == IDLESU * (1 - Pp * Palpha) + CW1 * (1 - P2st * (Piap * (Pii - 1) + 1)) +
        CW11 * (Piap * (1 - Pii)) + CW2 * (1 - Pbeta) + TXSU * (1 - Pp * Pii * Pgamma) &&
      CW1 == IDLESU * (Pp * Palpha) + TXSU * (Pp * Pii * Pgamma) &&
      CW11 == CW1 * (P2st * (1 - Piap)) + CW11 * (1 - Piap) &&
      CW2 == CW1 * (Pii * Piap * P2st) + CW11 * (Piap * Pii) &&
      TXSU == CW2 * Pbeta &&
      IDLESU + CW1 + CW11 + CW2 + TXSU == 1,
     {IDLESU, CW1, CW11, CW2, TXSU}] // FullSimplify;

solAP = Solve[IDLEAP == IDLEAP * (1 - Piap) + Pt1 * (1 - Psu1) * (1 - Piap) +
        Pt2 * (1 - Psu2) * (1 - Piap) + Pt2 * Psu2 * (1 - Piap) && Pt1 ==
       IDLEAP * Piap + Pt1 * Piap * (1 - Psu1) + Pt2 * Piap * (1 - Psu2) + Pt2 * Psu2 * Piap &&
      Pt11 == Pt1 * Psu1 * (1 - Piap) + Pt11 * (1 - Piap) &&
      Pt2 == Pt1 * Piap * Psu1 + Pt11 * Piap &&
      IDLEAP + Pt1 + Pt11 + Pt2 + ((cw2bAvg * Pt2 * Psu2) / Piap) == 1,
     {IDLEAP, Pt1, Pt11, Pt2}] // FullSimplify;

TXAP = ((cw2bAvg * (Pt2 /. solAP[[1]]) * Psu2) / Piap);

n = 2;
cw1 = 3;
cw2 = 5;
Piap = 0.5;
Pii = 0.5;

Pp = 1;

kmax = 20;
queueSize = 50;
```

```
n1Dist[n_, pii_, n1_] := If[pii == 1, Binomial[n, n1] * 0.9999^n1 * (1 - 0.99999)^(n - n1),
    Binomial[n, n1] * pii^n1 * (1 - pii)^(n - n1)];
n1Avg = Sum[n1 * n1Dist[n, Pii, n1], {n1, 0, n}];

n2naopondDist[n1_, n2_, cw1_] :=
  If[(1 / cw1) == 1, (Binomial[n1, n2] * 0.9999^(n2) * (1 - 0.9999)^(n1 - n2)),
   (Binomial[n1, n2] * (1 / cw1)^(n2) * (1 - (1 / cw1))^(n1 - n2))];
(*n2naopondAvg=N[Sum[n2*n2naopondDist[n1Avg,n2,cw1],{n2,0,n}]];*)

n2Dist[n_, pii_, cw1_, n2_] :=
  Sum[n1Dist[n, pii, n1] * n2naopondDist[n1, n2, cw1], {n1, 1, n}] /
   (1 - n1Dist[n, pii, 0]);
n2Avg = Sum[n2 * n2Dist[n, Pii, cw1, n2], {n2, 0, n}];

n2aDist[n_, pii_, cw1_, n2a_] :=
  Sum[n2Dist[n, pii, cw1, n2] * Binomial[n2, n2a] * pii^n2a * (1 - pii)^(n2 - n2a),
    {n2, 1, n}] / (1 - n2Dist[n, pii, cw1, 0]);
n2aAvg = Sum[n2a * n2aDist[n, Pii, cw1, n2a], {n2a, 0, n}];

Psu1 = 1 - n1Dist[n, Pii, 0];
Psu2 = 1 - n2aDist[n, Pii, cw1, 0];
P2st = (1 / cw1);
cw2bDist[n_, pii_, cw1_, cw2_, cw2b_] :=
  Sum[n2aDist[n, pii, cw1, n2a] * (1 / (cw2^n2a)) * Binomial[cw2, cw2b] *
     Sum[(-1)^tt * Binomial[cw2b, tt] * (cw2b - tt)^n2a, {tt, 0, cw2b - 1}],
    {n2a, 1, n}] / (1 - n2aDist[n, pii, cw1, 0]);
cw2bAvg = Sum[cw2b * cw2bDist[n, Pii, cw1, cw2, cw2b], {cw2b, 0, cw2}];

temposAPtoT1[cw_, cwdev_, piap_, k_] :=
  If[k ≤ 0, 0, PDF[NegativeBinomialDistribution[cw, piap] , k - cwdev]];
(*temposAPtoT1Avg=Sum[k*temposAPtoT1[cw2+1,cw2+1,Piap,k],{k,0,kmax}];*)

temposAPT2toT1pond[cw1_, cw2_, n_, pii_, piap_, k_] :=
  Sum[cw2bDist[n, pii, cw1, cw2, tt] * temposAPtoT1[tt + 1, tt + 1, piap, k], {tt, 1, cw2}];
temposAPT2toT1pondAvg =
  Sum[k * temposAPT2toT1pond[cw1, cw2, n, Pii, Piap, k], {k, 0, kmax}];

temposSUtoTX[cw2b_, pii_, k_] :=
  If[pii == 1, Sum[(1 / cw2b) * (1 - 0.999)^(k - v) * 0.999^(v) * Binomial[k - 1, v - 1],
    {v, 1, Min[k, cw2b]}], Sum[
    (1 / cw2b) * (1 - pii)^(k - v) * pii^(v) * Binomial[k - 1, v - 1], {v, 1, Min[k, cw2b]}]];

(*temposSUCW2toTXpond[cw1_,cw2_,n_,pii_,k_]:=
 Sum[cw2bDist[n,pii,cw1,cw2,tt]*temposSUtoTX[tt,pii,k],{tt,1,cw2}];
temposSUCW2toTXpondAvg=Sum[k*temposSUCW2toTXpond[cw1,cw2,n,Pii,k],{k,0,kmax}];*)

Pbeta =
  Sum[cw2bDist[n, Pii, cw1, cw2, cw2b] * Sum[temposAPtoT1[cw2b + 1, cw2b + 1, Piap, k] *
      Sum[temposSUtoTX[cw2b, Pii, l], {l, 0, k - 1}], {k, 1, kmax}], {cw2b, 1, cw2}];

Pgamma =
  Sum[cw2bDist[n, Pii, cw1, cw2, cw2b] * Sum[temposAPtoT1[cw2b + 1, cw2b + 1, Piap, k] *
       temposSUtoTX[cw2b, Pii, k - 1], {k, 1, kmax}], {cw2b, 1, cw2}] / Pbeta;

temposAPT1toT1pondAvg = (1 - Psu1) * Sum[k * temposAPtoT1[1, 1, Piap, k], {k, 0, kmax}] +
   Sum[k * temposAPtoT1[2, 2, Piap, k], {k, 0, kmax}] * Psu1 * (1 - Psu2) +
   (Sum[k * temposAPtoT1[1, 1, Piap, k], {k, 0, kmax}] +
      Sum[k * temposAPT2toT1pond[cw1, cw2, n, Pii, Piap, k], {k, 0, kmax}]) * Psu1 * Psu2;

ProbSUTxEmk[cw1_, cw2_, n_, pii_, piap_, k_] :=
  Sum[cw2bDist[n, pii, cw1, cw2, cw2b] * (temposSUtoTX[cw2b, pii, k] * Sum[temposAPtoT1[
        cw2b + 1, cw2b + 1, piap, i + 1], {i, k, kmax}]), {cw2b, 1, cw2}] / Pbeta;
ProbSUTxEmkAvg = Sum[k * ProbSUTxEmk[cw1, cw2, n, Pii, Piap, k], {k, 1, kmax}];
```

```
AvgTemposIdleCW1 =
  Sum[(k * temposAPT1toT1pondAvg - 1) * Pii * (1 - Pii) ^ (k - 1), {k, 1, kmax}];
AvgTemposIdleCW11 = Sum[(temposAPT1toT1pondAvg * k +
      Sum[k * temposAPtoT1[1, 1, Piap, k], {k, 0, kmax}] * (1 - Psu2) +
      (temposAPT2toT1pondAvg + 1) * Psu2) * Pii * (1 - Pii) ^ (k), {k, 0, kmax}];
AvgTemposIdleCW2 = Sum[(temposAPT1toT1pondAvg * k + temposAPT2toT1pondAvg) *
    Pii * (1 - Pii) ^ k, {k, 0, kmax}];
AvgTemposIdleTXSU = Sum[(temposAPT1toT1pondAvg * k + temposAPT2toT1pondAvg -
      ProbSUTxEmkAvg) * Pii * (1 - Pii) ^ k, {k, 0, kmax}];

Palpha = 1 / (AvgTemposIdleCW1 *
      (1 - (P2st * Piap * (1 - Piap + Pii + Pbeta * Pgamma * (-2 + Piap) * Pii ^ 2))) +
     AvgTemposIdleCW11 * (P2st * (1 - Piap) * Piap * (1 - Pii)) + AvgTemposIdleCW2 *
      ((Piap * Pii + (1 - Piap) * Piap * Pii) * P2st * (1 - Pbeta)) + AvgTemposIdleTXSU *
      ((Piap * Pii + (1 - Piap) * Piap * Pii) * P2st * Pbeta * (1 - Pii * Pgamma)));


IDLESU = (IDLESU /. solSU[[1]]);
CW1 = (CW1 /. solSU[[1]]);
CW11 = (CW11 /. solSU[[1]]);
CW2 = (CW2 /. solSU[[1]]);
TXSU = (TXSU /. solSU[[1]]);

(*Prof*)
pidle = (1 - TXSU) ^ n;
psucc = n * TXSU * (1 - TXSU) ^ (n - 1);
pcol = 1 - pidle - psucc;
pgood = psucc * Piap;

(*Miguel*)
TcicloAP = 1 / ((Pt2 /. solAP[[1]]) * Psu2);
TcicloSU = 1 / TXSU;

avgsuscicloap = (TcicloAP / TcicloSU) * n;
avgt2t1 = TcicloAP * TXAP;
tausu = 1 / avgt2t1;
idleporcicloap = avgt2t1 * (1 - tausu) ^ avgsuscicloap;
S = (avgt2t1 - idleporcicloap) / TcicloAP;
G = ((avgt2t1 - idleporcicloap) * Piap) / TcicloAP;

Print["n1: ", N[n1Avg], " n2: ", N[n2Avg], " n2a: ",
  N[n2aAvg], " P2st: ", N[P2st], " Psu1: ", N[Psu1], " Psu2: ",
  N[Psu2], " Pbeta: ", N[Pbeta], " Pgamma: ", N[Pgamma]];
Print["AP: IDLE: ", N[IDLEAP /. solAP[[1]]], " Pt1: ", N[Pt1 /. solAP[[1]]],
 " Pt11: ", N[Pt11 /. solAP[[1]]], " Pt2: ", N[Pt2 /. solAP[[1]]],
 " TX: ", N[TXAP], " SOMA: ", N[(IDLEAP /. solAP[[1]]) +
    (Pt1 /. solAP[[1]]) + (Pt11 /. solAP[[1]]) + (Pt2 /. solAP[[1]]) + TXAP]]
Print["SU: IDLE: ", N[IDLESU], " CW1: ", N[CW1], " CW11: ", N[CW11], " CW2: ",
  N[CW2], " TX: ", N[TXSU], " SOMA: ", N[IDLESU + CW1 + CW11 + CW2 + TXSU]];
Print["\nIDLE  -> IDLE:   ", N[1 - Pp * Palpha]];
Print["IDLE  -> CW1:    ", N[Pp * Palpha]];
Print["CW1   -> CW11:   ", N[P2st * (1 - Piap)]];
Print["CW1   -> CW2:    ", N[Pii * Piap * P2st]];
Print["CW1   -> IDLE:   ", N[1 - P2st * (Piap * (Pii - 1) + 1)]];
Print["CW11  -> CW2:    ", N[Piap * Pii]];
Print["CW11  -> CW11:   ", N[1 - Piap]];
Print["CW11  -> IDLE:   ", N[Piap * (1 - Pii)]];
Print["CW2   -> IDLE:   ", N[1 - Pbeta]];
Print["CW2   -> TX:     ", N[Pbeta]];
Print["TX    -> IDLE:   ", N[1 - Pp * Pii * Pgamma]];
Print["TX    -> CW1:    ", N[Pp * Pii * Pgamma]];
Print["Prob Idle: ", N[pidle], " Prob Succ: ",
  N[psucc], " Prob Coll: ", N[pcol], " Prob Good: ", N[pgood]];
```

```
Print["Prob Succ Mig: ", N[S], " Prob Good Mig: ", N[G]];

ClearAll["Global`*"]

solSU = Solve[IDLESU == IDLESU * (1 - Pp * Palpha) + CW1 * (1 - P2st * (Piap * (Pii - 1) + 1)) +
        CW11 * (Piap * (1 - Pii)) + CW2 * (1 - Pbeta) + TXSU * (1 - Pp * Pii * Pgamma) &&
      CW1 == IDLESU * (Pp * Palpha) + TXSU * (Pp * Pii * Pgamma) &&
      CW11 == CW1 * (P2st * (1 - Piap)) + CW11 * (1 - Piap) &&
      CW2 == CW1 * (Pii * Piap * P2st) + CW11 * (Piap * Pii) &&
      TXSU == CW2 * Pbeta &&
      IDLESU + CW1 + CW11 + CW2 + TXSU == 1,
     {IDLESU, CW1, CW11, CW2, TXSU}] // FullSimplify;

solAP = Solve[IDLEAP == IDLEAP * (1 - Piap) + Pt1 * (1 - Psu1) * (1 - Piap) +
         Pt2 * (1 - Psu2) * (1 - Piap) + Pt2 * Psu2 * (1 - Piap) && Pt1 ==
       IDLEAP * Piap + Pt1 * Piap * (1 - Psu1) + Pt2 * Piap * (1 - Psu2) + Pt2 * Psu2 * Piap &&
      Pt11 == Pt1 * Psu1 * (1 - Piap) + Pt11 * (1 - Piap) &&
      Pt2 == Pt1 * Piap * Psu1 + Pt11 * Piap &&
      IDLEAP + Pt1 + Pt11 + Pt2 + ((cw2bAvg * Pt2 * Psu2) / Piap) == 1,
     {IDLEAP, Pt1, Pt11, Pt2}] // FullSimplify;

TXAP = ((cw2bAvg * (Pt2 /. solAP[[1]]) * Psu2) / Piap);

n1Dist[n_, pii_, n1_] := If[pii == 1, Binomial[n, n1] * 0.9999^n1 * (1 - 0.99999)^(n - n1),
    Binomial[n, n1] * pii^n1 * (1 - pii)^(n - n1)];

n2naopondDist[n1_, n2_, cw1_] :=
  If[(1 / cw1) == 1, (Binomial[n1, n2] * 0.9999^(n2) * (1 - 0.9999)^(n1 - n2)),
   (Binomial[n1, n2] * (1 / cw1)^(n2) * (1 - (1 / cw1))^(n1 - n2))];

n2Dist[n_, pii_, cw1_, n2_] :=
  Sum[n1Dist[n, pii, n1] * n2naopondDist[n1, n2, cw1], {n1, 1, n}] /
   (1 - n1Dist[n, pii, 0]);

n2aDist[n_, pii_, cw1_, n2a_] :=
  Sum[n2Dist[n, pii, cw1, n2] * Binomial[n2, n2a] * pii^n2a * (1 - pii)^(n2 - n2a),
    {n2, 1, n}] / (1 - n2Dist[n, pii, cw1, 0]);

cw2bDist[n_, pii_, cw1_, cw2_, cw2b_] :=
  Sum[n2aDist[n, pii, cw1, n2a] * (1 / (cw2^n2a)) * Binomial[cw2, cw2b] *
      Sum[(-1)^tt * Binomial[cw2b, tt] * (cw2b - tt)^n2a, {tt, 0, cw2b - 1}],
    {n2a, 1, n}] / (1 - n2aDist[n, pii, cw1, 0]);

temposAPtoT1[cw_, cwdev_, piap_, k_] :=
  If[k ≤ 0, 0, PDF[NegativeBinomialDistribution[cw, piap] , k - cwdev]];

temposAPT2toT1pond[cw1_, cw2_, n_, pii_, piap_, k_] :=
  Sum[cw2bDist[n, pii, cw1, cw2, tt] * temposAPtoT1[tt + 1, tt + 1, piap, k], {tt, 1, cw2}];

temposSUtoTX[cw2b_, pii_, k_] :=
  If[pii == 1, Sum[(1 / cw2b) * (1 - 0.999)^(k - v) * 0.999^(v) * Binomial[k - 1, v - 1],
    {v, 1, Min[k, cw2b]}], Sum[
    (1 / cw2b) * (1 - pii)^(k - v) * pii^(v) * Binomial[k - 1, v - 1], {v, 1, Min[k, cw2b]}]];

ProbSUTxEmk[cw1_, cw2_, n_, pii_, piap_, k_] :=
  Sum[cw2bDist[n, pii, cw1, cw2, cw2b] * (temposSUtoTX[cw2b, pii, k] * Sum[temposAPtoT1[
          cw2b + 1, cw2b + 1, piap, i + 1], {i, k, kmax}]), {cw2b, 1, cw2}] / Pbeta;


For[cw1 = 1, cw1 ≤ 3, cw1 = cw1 + 1,
 For[cw2 = 6, cw2 ≤ 6, cw2 = cw2 + 1,
   For[piap=0.3, piap ≤ 0.7, piap=piap+0.2,
    For[pii=piap-0.1, pii≤piap+0.1, pii=pii+0.1,
       For[n = 0, n < 51, n = n + 3;
```

```
Piap = piap;
Pii = pii;

Pp = 1;

kmax = 20;

queueSize = 50;


n1Avg = Sum[n1 * n1Dist[n, Pii, n1], {n1, 0, n}];

n2Avg = Sum[n2 * n2Dist[n, Pii, cw1, n2], {n2, 0, n}];

n2aAvg = Sum[n2a * n2aDist[n, Pii, cw1, n2a], {n2a, 0, n}];

Psu1 = 1 - n1Dist[n, Pii, 0];

Psu2 = 1 - n2aDist[n, Pii, cw1, 0];

P2st = (1 / cw1);

cw2bAvg = Sum[cw2b * cw2bDist[n, Pii, cw1, cw2, cw2b], {cw2b, 0, cw2}];

temposAPT2toT1pondAvg =
 Sum[k * temposAPT2toT1pond[cw1, cw2, n, Pii, Piap, k], {k, 0, kmax}];

Pbeta = Sum[cw2bDist[n, Pii, cw1, cw2, cw2b] *
    Sum[temposAPtoT1[cw2b + 1, cw2b + 1, Piap, k] * Sum[temposSUtoTX[cw2b, Pii, l],
        {l, 0, k - 1}], {k, 1, kmax}], {cw2b, 1, cw2}];

Pgamma = Sum[cw2bDist[n, Pii, cw1, cw2, cw2b] *
     Sum[temposAPtoT1[cw2b + 1, cw2b + 1, Piap, k] * temposSUtoTX[cw2b, Pii, k - 1],
      {k, 1, kmax}], {cw2b, 1, cw2}] / Pbeta;

temposAPT1toT1pondAvg =
 (1 - Psu1) * Sum[k * temposAPtoT1[1, 1, Piap, k], {k, 0, kmax}] +
  Sum[k * temposAPtoT1[2, 2, Piap, k], {k, 0, kmax}] * Psu1 * (1 - Psu2) +
  (Sum[k * temposAPtoT1[1, 1, Piap, k], {k, 0, kmax}] + Sum[k * temposAPT2toT1pond[
        cw1, cw2, n, Pii, Piap, k], {k, 0, kmax}]) * Psu1 * Psu2;

ProbSUTxEmkAvg = Sum[k * ProbSUTxEmk[cw1, cw2, n, Pii, Piap, k], {k, 1, kmax}];

AvgTemposIdleCW1 =
 Sum[(k * temposAPT1toT1pondAvg - 1) * Pii * (1 - Pii) ^ (k - 1), {k, 1, kmax}];

AvgTemposIdleCW11 = Sum[(temposAPT1toT1pondAvg * k +
      Sum[k * temposAPtoT1[1, 1, Piap, k], {k, 0, kmax}] * (1 - Psu2) +
      (temposAPT2toT1pondAvg + 1) * Psu2) * Pii * (1 - Pii) ^ (k), {k, 0, kmax}];

AvgTemposIdleCW2 = Sum[(temposAPT1toT1pondAvg * k + temposAPT2toT1pondAvg) *
    Pii * (1 - Pii) ^k, {k, 0, kmax}];

AvgTemposIdleTXSU = Sum[(temposAPT1toT1pondAvg * k + temposAPT2toT1pondAvg -
      ProbSUTxEmkAvg) * Pii * (1 - Pii) ^k, {k, 0, kmax}];

Palpha = 1 / (AvgTemposIdleCW1 *
      (1 - (P2st * Piap * (1 - Piap + Pii + Pbeta * Pgamma * (-2 + Piap) * Pii ^ 2))) +
     AvgTemposIdleCW11 * (P2st * (1 - Piap) * Piap * (1 - Pii)) + AvgTemposIdleCW2 *
      ((Piap * Pii + (1 - Piap) * Piap * Pii) * P2st * (1 - Pbeta)) + AvgTemposIdleTXSU *
      ((Piap * Pii + (1 - Piap) * Piap * Pii) * P2st * Pbeta * (1 - Pii * Pgamma)));
```

```
        IDLESU = (IDLESU /. solSU[[1]]);
        CW1 = (CW1 /. solSU[[1]]);
        CW11 = (CW11 /. solSU[[1]]);
        CW2 = (CW2 /. solSU[[1]]);
        TXSU = (TXSU /. solSU[[1]]);


        (*Miguel*)
        TcicloAP = 1 / ((Pt2 /. solAP[[1]]) * Psu2);
        TcicloSU = 1 / TXSU;

        avgsuscicloap = (TcicloAP / TcicloSU) * n;
        avgt2t1 = TcicloAP * TXAP;
        tausu = 1 / avgt2t1;
        idleporcicloap = avgt2t1 * (1 - tausu) ^ avgsuscicloap;
        S = (avgt2t1 - idleporcicloap) / TcicloAP;
        G = ((avgt2t1 - idleporcicloap) * Piap) / TcicloAP;


        G >>> "C:\\Users\\Miguel Luis\\Documents\\My
            Dropbox\\Doutoramento\\Artigos\\SingleRadio\\heterogeneous_tons
            \\validation\\throughput\\theoretical.txt";

        TXSU >>> "C:\\Users\\Miguel Luis\\Documents\\My
            Dropbox\\Doutoramento\\Artigos\\SingleRadio\\heterogeneous_tons
            \\validation\\steady_txsu\\theoretical.txt";

       ];


       {{{{}}}} >>> "C:\\Users\\Miguel Luis\\Documents\\My
          Dropbox\\Doutoramento\\Artigos\\SingleRadio\\heterogeneous_tons\\
          validation\\throughput\\theoretical.txt";

       {{{{}}}} >>> "C:\\Users\\Miguel Luis\\Documents\\My
          Dropbox\\Doutoramento\\Artigos\\SingleRadio\\heterogeneous_tons\\
          validation\\steady_txsu\\theoretical.txt";

      ];
    ];

  ];

]
```

```
ClearAll["Global`*"]

solSU = Solve[IDLESU == IDLESU * (1 - Pp * Palpha) + CW1 * (1 - P2st * (Piap * (Pii - 1) + 1)) +
       CW11 * (Piap * (1 - Pii)) + CW2 * (1 - Pbeta) + TXSU * (1 - Pp * Pii * Pgamma) &&
     CW1 == IDLESU * (Pp * Palpha) + TXSU * (Pp * Pii * Pgamma) &&
     CW11 == CW1 * (P2st * (1 - Piap)) + CW11 * (1 - Piap) &&
     CW2 == CW1 * (Pii * Piap * P2st) + CW11 * (Piap * Pii) &&
     TXSU == CW2 * Pbeta &&
     IDLESU + CW1 + CW11 + CW2 + TXSU == 1,
    {IDLESU, CW1, CW11, CW2, TXSU}] // FullSimplify

solAP = Solve[IDLEAP == IDLEAP * (1 - Piap) + Pt1 * (1 - Psu1) * (1 - Piap) +
       Pt2 * (1 - Psu2) * (1 - Piap) + Pt2 * Psu2 * (1 - Piap) && Pt1 ==
      IDLEAP * Piap + Pt1 * Piap * (1 - Psu1) + Pt2 * Piap * (1 - Psu2) + Pt2 * Psu2 * Piap &&
     Pt11 == Pt1 * Psu1 * (1 - Piap) + Pt11 * (1 - Piap) &&
     Pt2 == Pt1 * Piap * Psu1 + Pt11 * Piap &&
     IDLEAP + Pt1 + Pt11 + Pt2 + ((cw2bAvg * Pt2 * Psu2) / Piap) == 1,
    {IDLEAP, Pt1, Pt11, Pt2}] // FullSimplify

TXAP = ((cw2bAvg * (Pt2 /. solAP[[1]]) * Psu2) / Piap);

SS = 1 / (TXSU /. solSU[[1]])

D[SS, Piap]
```