```
fp[t_, q1_, q2_, b1_, b2_, at_, aq1_, aq2_, ab1_, ab2_]=0
afp=Compile[{t,q1,q2,b1,b2,at,aq1,aq2,ab1,ab2},
 fp[t,q1,q2,b1,b2,at,aq1,aq2,ab1,ab2]]
c=Table[afp[tp[i],q1[i],q2[i],b1[i],b2[i],tp[j],q1[j],q2[j],b1[j],b2[j]],
 {i,0,Ix[1,Q-1,Q-1,B-1,B-1]},{j,0,Ix[1,Q-1,Q-1,B-1,B-1]}]
(*d=Table[{i,Sum[c[[i,j]], {j, 1, Ix[1,Q-1,Q-1,B-1,B-1]+1}]},
 {i, 1, Ix[1,Q-1,Q-1,B-1,B-1]+1}]*)
NonExit=Table[Apply[And,Table[c[[i,j]]==0, {i,1,Ix[1,Q-1,Q-1,B-1,B-1]+1}]],
 {j,1,Ix[1,Q-1,Q-1,B-1,B-1]+1}]
(*ZeroCol=Table[First[Rest[d[[j]]]]==0, {j,1,Ix[1,Q-1,Q-1,B-1,B-1]+1}]
Comp=Table[NonExit[[i]]==ZeroCol[[i]], {i, 1,Ix[1,Q-1,Q-1,B-1,B-1]+1}]*)
NonIx=Map[First,Position[Table[If[NonExit[[i]]==True,i,,],{i,1,Length[NonExit]}],Null]]
ft[1,qq1_,qq2_,bb1_,bb2_]:=1+a
ft[0,0,0,0,0]:=1/(l1+l2)
ft[0,qq1_,qq2_,bb1_,bb2_]:=a*2^Min[bb1,bb2]
tt=Table[ft[tp[i],q1[i],q2[i],b1[i],b2[i]],{i,0,Ix[1,Q-1,Q-1,B-1,B-1]}]
ctt=Part[tt,NonIx]
fu[1,qq1_,qq2_,bb1_,bb2_]:=1
fu[0,qq1_,qq2_,bb1_,bb2_]:=0
tu:=Table[fu[tp[i],q1[i],q2[i],b1[i],b2[i]],{i,0,Ix[1,Q-1,Q-1,B-1,B-1]}]
ctu=Part[tu,NonIx]
fq[aa_,qq1_,qq2_,bb1_,bb2_]:=qq1+qq2
tq=Table[fq[tp[i],q1[i],q2[i],b1[i],b2[i]],{i,0,Ix[1,Q-1,Q-1,B-1,B-1]}]
ctq=Part[tq,NonIx]
e=Part[c,NonIx,NonIx]
(*f=Table[{i,Sum[e[[i,j]], {j,1,Length[e]}]},{i,1,Length[e]}]
CheckOK=Count[NonExit,False]==Length[e]
NonExCheck=Table[Apply[And,Table[e[[i,j]]==0,{i,1,Length[e]}]],{j,1,Length[e]}]
*)
e1=IdentityMatrix[Length[e]]-Transpose[e]
solution=LinearSolve[Append[Delete[e1,{Length[e]}],Table[1,{i,1,Length[e]}]], Table[I-
f[i==Length[e],1,0],{i,1,Length[e]}]].MatrixPower[e,10]
(*Solution=Expandvec[solution]*)
(*CMP[x_]:=GrayLevel[1-x]
CMT[x_]:=GrayLevel[If[x==0,1,0]]*)
IxFil=Part[Table[Px[i-1],{i,1,Length[c]}],NonIx]
(*pldenz=ListDensityPlot[e,ColorFunction:>(CMP[#]&)]*)
IxLst=Table[{IxFil[[i]],solution[[i]]},{i,1,Length[solution]}]
qlen=solution.ctq
(*Little’s law T=N/lambda*)
Nsys=qlen+solution.ctu
Tsys=Nsys/(l1+l2)
Tsys2=solution.ctu/(l1+l2)
Tsys3=qlen/(l1+l2)
throughput=solution.ctu/solution.ctt
```

```
fp[1, l_/;l>0, m_/;m>0, q_/;q>0, 0, 0, n_/;n>0, k_/;k>0, p_, 1]:=
 1/2^q Po1[1,n-l] Po2[1,k-m]/;n>=l&&k>=m&&n<Q-1&&k<Q-1&&p==Mod[q+1,B]
fp[1, l_/;l>0, m_/;m>0, q_/;q>0, 0, 0, Q-1, k_/;k>0, p_, 1]:=
 1/2^q (1-Sum[Po1[1,n],{n,0,Q-l-2}]) *
 Po2[1,k-m]/;k>=m&&k<Q-1&&p==Mod[q+1,B]
fp[1, l_/;l>0, m_/;m>0, q_/;q>0, 0, 0, n_/;n>0, Q-1, p_, 1]:=
 1/2^q Po1[1,n-l] *
 (1-Sum[Po2[1,k],{k,0,Q-m-2}])/;n>=l&&n<Q-1&&p==Mod[q+1,B]
fp[1, l_/;l>0, m_/;m>0, q_/;q>0, 0, 0, Q-1, Q-1, p_, 1]:=
 1/2^q (1-Sum[Po1[1,n],{n,0,Q-l-2}]) *
 (1-Sum[Po2[1,k],{k,0,Q-m-2}])/;p==Mod[q+1,B]

fp[1, l_/;l>0, m_/;m>0,0, q_/;q>0, 1, n_, k_/;k>0, 0, p_/;p>0]:=
 (1-1/2^q)Po1[1,n-l+1] Po2[1,k-m]/;n>=l-1&&k>=m&&n<Q-1&&k<Q-1&&p==q
fp[1, l_/;l>0, m_/;m>0, 0, q_/;q>0, 1, Q-1, k_/;k>0, 0, p_/;p>0]:=
 (1-1/2^q) (1-Sum[Po1[1,n],{n,0,Q-l-1}]) *
 Po2[1,k-m]/;k>=m&&k<Q-1&&p==q
fp[1, l_/;l>0, m_/;m>0, 0, q_/;q>0, 1, n_, Q-1, 0, p_/;p>0]:=
 (1-1/2^q)Po1[1,n-l+1] *
 (1-Sum[Po2[1,k],{k,0,Q-m-2}])/;n>=l-1&&n<Q-1&&p==q
fp[1, l_/;l>0, m_/;m>0, 0, q_/;q>0, 1, Q-1, Q-1, 0, p_/;p>0]:=
 (1-1/2^q)(1-Sum[Po1[1,n],{n,0,Q-l-1}]) *
 (1-Sum[Po2[1,k],{k,0,Q-m-2}])/;p==q

fp[1, l_/;l>0, m_/;m>0, 0, q_/;q>0, 0, n_/;n>0, k_/;k>0, 1,p_]:=
 1/2^q Po1[1,n-l] Po2[1,k-m]/;n>=l&&k>=m&&n<Q-1&&k<Q-1&&p==Mod[q+1,B]
fp[1, l_/;l>0, m_/;m>0, 0, q_/;q>0, 0, Q-1, k_/;k>0, 1,p_]:=
 1/2^q (1-Sum[Po1[1,n],{n,0,Q-l-2}]) *
 Po2[1,k-m]/;k>=m&&k<Q-1&&p==Mod[q+1,B]
fp[1, l_/;l>0, m_/;m>0, 0, q_/;q>0, 0, n_/;n>0, Q-1, 1,p_]:=
 1/2^q Po1[1,n-l] *
 (1-Sum[Po2[1,k],{k,0,Q-m-2}])/;n>=l&&n<Q-1&&p==Mod[q+1,B]
fp[1, l_/;l>0, m_/;m>0, 0, q_/;q>0, 0, Q-1, Q-1, 1,p_]:=
 1/2^q (1-Sum[Po1[1,n],{n,0,Q-l-2}]) *
 (1-Sum[Po2[1,k],{k,0,Q-m-2}])/;p==Mod[q+1,B]

(*fp[1,l_/;l>0,0,q_,0, 0,l_+j_/;j>-1,k, Mod[q_+1, B],1]=
 l2^k l1^j Exp[-l2 k -l2 j]*)
(*Possible error here- train of packets -the variable t should
be extended to have the chaining of privileged packets described, like:
fp[1,0,l_/;l>0,0,q_, 2,k_,m_,0,n_]:=l1^k l2^(m-l) *
 Exp[-l1 k -l2 (m-l)] * Max[0,(2^q-1/a-1)/2^q] *Max[0,(2^q-2/a-2)/2^q]
 /; (m>l-1 &&n==Mod[q+1,B] && !(m==l && k==0))
*)
```

```
 (1-Min[1,(1+1/a)/2^q]) *
 (1-Min[1,(2+2/a)/2^q]) /; (m>=l &&n==Mod[q+1,B]&&m<Q-1)
fp[1,0,l_/;l>0,0,q_/;q>0, 1,k_/;k>0,Q-1,0,n_]:=Po1[1,k] *
 (1-Sum[Po2[1,m], {m, 0, Q-l-2}]) *
 (1-Min[1,(1+1/a)/2^q]) *
 (1-Min[1,(2+2/a)/2^q]) /; n==Mod[q+1,B]

fp[1,l_/;l>0,0,q_/;q>0,0, 1,k_,0,0,0]:=Exp[-l1-l2] /; k==l-1
fp[1,Q-1,0,q_,0, 1,Q-1,0,0,0]:=Exp[-l2](1-Exp[-l1])
fp[1,l_/;l>0,0,q_/;q>0,0, 1,k_/;k>0,0,0,0]:=Po2[1,0] *
 Po1[1,k-l+1] /;k>=l && k<Q-1
fp[1,l_/;l>0,0,q_/;q>0,0, 1,Q-1,0,0,0]:=Po2[1,0]*
 (1-Sum[Po1[1,k],{k,0,Q-l-1}])
fp[1,l_/;l>0,0,q_/;q>0,0, 0,m_,k_/;k>0,n_,1]:=Po2[1,k] Po1[1,m-l] *
 Min[1,(1+1/a)/2^q] /;(m>=l&&n==Mod[q+1,B]&&m<Q-1)

fp[1,l_/;l>0,0,q_/;q>0,0, 0,Q-1,k_/;k>0,n_,1]:=Po2[1,k] *
 (1-Sum[Po1[1,m], {m, 0, Q-l-2}]) *
 Min[1,(1+1/a)/2^q] /;n==Mod[q+1,B]

fp[1,l_/;l>0,0,q_/;q>0,0, 1,m_,k_/;k>0,0,0]:=Po2[1,k] Po1[1,m-l] *
 (1-Min[1,(1+1/a)/2^q]) *
 Min[1,(2+2/a)/2^q] /; (m>=l &&m<Q-1)
fp[1,l_/;l>0,0,q_/;q>0,0, 1,Q-1,k_/;k>0,0,0]:=Po2[1,k] *
 (1-Sum[Po1[1,m], {m, 0, Q-l-2}]) *
 (1-Min[1,(1+1/a)/2^q])*Min[1,(2+2/a)/2^q]

fp[1,l_/;l>0,0,q_/;q>0,0, 1,m_,k_/;k>0,n_,0]:=Po2[1,k] Po1[1,m-l] *
 (1-Min[1,(1+1/a)/2^q]) *
 (1-Min[1,(2+2/a)/2^q]) /; (m>=l &&n==Mod[q+1,B]&&m<Q-1)
fp[1,l_/;l>0,0,q_/;q>0,0, 1,Q-1,k_/;k>0,n_,0]:=Po2[1,k] *
 (1-Sum[Po1[1,m], {m, 0, Q-l-2}]) *
 (1-Min[1,(1+1/a)/2^q]) *
 (1-Min[1,(2+2/a)/2^q]) /; n==Mod[q+1,B]

fp[1, l_/;l>0, m_/;m>0, q_/;q>0, 0, 1, n_/;n>0, k_, p_/;p>0, 0]:=
 (1-1/2^q)Po1[1,n-l] Po2[1,k-m+1]/;n>=l&&k>=m-1&&n<Q-1&&k<Q-1&&p==q
fp[1, l_/;l>0, m_/;m>0, q_/;q>0, 0, 1, Q-1, k_, p_/;p>0, 0]:=
 (1-1/2^q) (1-Sum[Po1[1,n],{n,0,Q-l-2}]) *
 Po2[1,k-m+1]/;k>=m-1&&k<Q-1&&p==q
fp[1, l_/;l>0, m_/;m>0, q_/;q>0, 0, 1, n_/;n>0, Q-1, p_/;p>0, 0]:=
 (1-1/2^q)Po1[1,n-l] *
 (1-Sum[Po2[1,k],{k,0,Q-m-1}])/;n>=l&&n<Q-1&&p==q
fp[1, l_/;l>0, m_/;m>0, q_/;q>0, 0, 1, Q-1, Q-1, p_/;p>0, 0]:=
 (1-1/2^q)(1-Sum[Po1[1,n],{n,0,Q-l-2}]) *
 (1-Sum[Po2[1,k],{k,0,Q-m-1}])/;p==q
```

```
fp[1,m_/;m>0,k_/;k>0,0,0, 0,l_/;l>0,n_/;n>0,1,1]:=Po1[1,l-m] *
 Po2[1,n-k]/;n>=k && n<Q-1 && l>=m && l<Q-1
fp[1,m_/;m>0,k_/;k>0,0,0, 0,Q-1,n_/;n>0,1,1]:=(1-Sum[Po1[1,l],{l,0,Q-m-2}]) *
 Po2[1,n-k]/;n>=k && n<Q-1
fp[1,m_/;m>0,k_/;k>0,0,0, 0,l_/;l>0,Q-1,1,1]:=(1-Sum[Po2[1,n],{n,0,Q-k-2}]) *
 Po1[1,l-m]/;l>=m && l<Q-1
fp[1,m_/;m>0,k_/;k>0,0,0, 0,Q-1,Q-1,1,1]:=(1-Sum[Po2[1,n],{n,0,Q-k-2}]) *
(1-Sum[Po1[1,l],{l,0,Q-m-2}])


fp[1,m_/;m>0,k_/;k>0,0,0, 0,l_/;l>0,n_/;n>0,1,1]:=Po1[1,l-m] *
 Po2[1,n-k]/; l>=m && n>=k&&l<Q-1&&n<Q-1
fp[1,m_/;m>0,k_/;k>0,0,0, 0,Q-1,n_/;n>0,1,1]:=Po2[1,n-k] *
(1- Sum[Po1[1,l],{l,0,Q-m-2}])/; n>=k &&n<Q-1
fp[1,m_/;m>0,k_/;k>0,0,0, 0,l_/;l>0,Q-1,1,1]:=Po1[1,l-m] *
(1- Sum[Po2[1,n],{n,0,Q-k-2}])/; l>=m&&l<Q-1
fp[1,m_/;m>0,k_/;k>0,0,0, 0,Q-1,Q-1,1,1]:=
 (1-Sum[Po1[1,l],{l,0,Q-m-2}]) *
 (1- Sum[Po2[1,n],{n,0,Q-k-2}])

(*fp[1,m_/;m>0,k_/;k>0,0,0, 0,l_/;l>0,n_/;n>0,1,1]:=(Exp[-l1 ]) *
 (l2 )^(n-k) Exp[- l2 (n-k)]/;m==l && n>k
fp[1,m_/;m>0,k_/;k>0,0,0, 0,l_/;l>0,n_/;n>0,1,1]:=(Exp[-l2 ]) *
 (l1 )^(l-m) Exp[- l1 (l-m)]/;n==k && l>m *)

fp[1,0,l_/;l>0,0,q_/;q>0, 1,0,k_,0,0]:=Exp[-l1-l2]/; k==l-1
fp[1,0,Q-1,0,q_, 1,0,Q-1,0,0]:=Exp[-l1](1-Exp[-l2])
fp[1,0,l_/;l>0,0,q_/;q>0, 1,0,k_/;k>0,0,0]:=Po1[1,0]*
 Po2[1,k-l+1]/;k>=l && k<Q-1
fp[1,0,l_/;l>0,0,q_/;q>0, 1,0,Q-1,0,0]:=Po1[1,0]*
 (1-Sum[Po2[1,k],{k,0,Q-l-1}])
fp[1,0,l_/;l>0,0,q_/;q>0, 0,k_/;k>0,m_,1,n_]:=Po1[1,k] Po2[1,m-l] *
 Min[1,(1+1/a)/2^q] /;(m>=l &&n==Mod[q+1,B]&&m<Q-1)

fp[1,0,l_/;l>0,0,q_/;q>0, 0,k_/;k>0,Q-1,1,n_]:=Po1[1,k] *
 (1-Sum[Po2[1,m], {m, 0, Q-l-2}]) *
 Min[1,(1+1/a)/2^q] /;n==Mod[q+1,B]

fp[1,0,l_/;l>0,0,q_/;q>0, 1,k_/;k>0,m_,0,0]:=Po1[1,k] Po2[1,m-l] *
 (1-Min[1,(1+1/a)/2^q]) *
 Min[1,(2+2/a)/2^q] /; (m>=l&&m<Q-1)
fp[1,0,l_/;l>0,0,q_/;q>0, 1,k_/;k>0,Q-1,0,0]:=Po1[1,k] *
 (1-Sum[Po2[1,m], {m, 0, Q-l-2}]) *
 (1-Min[1,(1+1/a)/2^q])*Min[1,(2+2/a)/2^q]

fp[1,0,l_/;l>0,0,q_/;q>0, 1,k_/;k>0,m_,0,n_]:=Po1[1,k] Po2[1,m-l] *
```

```
p1[l,n] /; o>=m && p>=k && o<Q-1 && p<Q-1 && r==Mod[l+1,B] && s==Mod[n+1,B]
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 0,o_,Q-1,r_,s_]:=
 Po1[a Ep[l,n],o-m](1-Sum[Po2[a Ep[l,n],p],{p,0,Q-k-2}]) *
 p1[l,n]/;o>=m && o<Q-1 && r==Mod[l+1,B] && s==Mod[n+1,B]
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 0,Q-1,p_,r_,s_]:=
 Po2[a Ep[l,n],p-k](1-Sum[Po1[a Ep[l,n],o],{o,0,Q-m-2}]) *
 p1[l,n]/;p>=k && p<Q-1 && r==Mod[l+1,B] && s==Mod[n+1,B]
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 0,Q-1,Q-1,r_,s_]:=
(1-Sum[Po2[a Ep[l,n],p],{p,0,Q-k-2}])(1-Sum[Po1[a Ep[l,n],o],{o,0,Q-m-2}]) *
 p1[l,n]/; r==Mod[l+1,B] && s==Mod[n+1,B]

fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 1,o_,p_,0,s_]:=
 Po1[a Ep[l,n],o-m+1] Po2[a Ep[l,n],p-k] *
 p2[l,n]/; o>=m-1 && p>=k && o<Q-1 && p<Q-1 && s==n
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 1,o_,Q-1,0,s_]:=
 Po1[a Ep[l,n],o-m+1](1-Sum[Po2[a Ep[l,n],p],{p,0,Q-k-2}]) *
 p2[l,n]/;o>=m-1 && o<Q-1 && s==n
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 1,Q-1,p_,0,s_]:=
 Po2[a Ep[l,n],p-k](1-Sum[Po1[a Ep[l,n],o],{o,0,Q-m-1}]) *
 p2[l,n]/;p>=k && p<Q-1 && s==n
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 1,Q-1,Q-1,0,s_]:=
(1-Sum[Po2[a Ep[l,n],p],{p,0,Q-k-2}])(1-Sum[Po1[a Ep[l,n],o],{o,0,Q-m-1}]) *
 p2[l,n]/; s==Mod[n+1,B]

fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 1,o_,p_,s_,0]:=
 Po1[a Ep[l,n],o-m] Po2[a Ep[l,n],p-k+1] *
 p2[n,l]/; o>=m && p>=k-1 && o<Q-1 && p<Q-1 && s==l
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 1,o_,Q-1,s_,0]:=
 Po1[a Ep[l,n],o-m](1-Sum[Po2[a Ep[l,n],p],{p,0,Q-k-1}]) *
 p2[n,l]/;o>=m && o<Q-1 && s==l
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 1,Q-1,p_,s_,0]:=
 Po2[a Ep[l,n],p-k+1](1-Sum[Po1[a Ep[l,n],o],{o,0,Q-m-2}]) *
 p2[n,l]/;p>=k-1 && p<Q-1 && s==l
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 1,Q-1,Q-1,s_,0]:=
(1-Sum[Po2[a Ep[l,n],p],{p,0,Q-k-1}])(1-Sum[Po1[a Ep[l,n],o],{o,0,Q-m-2}]) *
 p2[n,l]/; s==Mod[l+1,B]

fp[1,m_/;m>0,0,0,0, 1,n_,0,0,0]:=Po1[1,n-m+1] Po2[1,0]/;n>=m-1&&n<Q-1
fp[1,m_/;m>0,0,0,0, 1,Q-1,0,0,0]:=Po2[1,0](1-Sum[Po1[1,n],{n,0,Q-m-1}])
fp[1,m_/;m>0,0,0,0, 0,n_,k_/;k>0,1,1]:=Po1[1,n-m] Po2[1,k]/;n>=m&&n<Q-1
fp[1,m_/;m>0,0,0,0, 0,Q-1,k_/;k>0,1,1]:=Po2[1,k](1-Sum[Po1[1,n],{n,0,Q-m-2}])
fp[1,0,m_/;m>0,0,0, 1,0,n_,0,0]:=Po2[1,n-m+1] Po1[1,0]/;n>=m-1&&n<Q-1
fp[1,0,m_/;m>0,0,0, 1,0,Q-1,0,0]:=Po1[1,0](1-Sum[Po2[1,n],{n,0,Q-m-1}])
fp[1,0,m_/;m>0,0,0, 0,k_/;k>0,n_,1,1]:=Po2[1,n-m] Po1[1,k]/;n>=m&&n<Q-1
fp[1,0,m_/;m>0,0,0, 0,k_/;k>0,Q-1,1,1]:=Po1[1,k](1-Sum[Po2[1,n],{n,0,Q-m-2}])
```

```
fp[0,0,m_/;m>0,0,l_/;l>0, 0,n_/;n>0,k_,1,o_]:=Po2[a 2^(l-1),k-m] *
 1/2^l Po1[a,n]/;k>=m && o==Mod[l+1,B]
fp[0,0,m_/;m>0,0,l_/;l>0, 1,0,n_,0,0]:=Po1[2^(l-1) a,0] *
Po2[2^(l-1) a, n-m+1]/;n>=m-1 && n<Q-1
fp[0,0,m_/;m>0,0,l_/;l>0, 1,0,Q-1,0,0]:=Po1[2^(l-1) a,0] *
(1-Sum[Po2[2^(l-1) a, n],{n,0,Q-m-1}])
fp[0,0,m_/;m>0,0,l_/;l>0, 1,n_,o_,0,p_]:=Po1[2^(l-1) a,n+1] *
Po2[2^(l-1) a, o-m]/; o>=m && p==l&& o<Q-1
fp[0,0,m_/;m>0,0,l_/;l>0, 1,n_,Q-1,0,p_]:=Po1[2^(l-1) a,n+1] *
(1-Sum[Po2[2^(l-1) a, o],{o,0,Q-m-2}])/; p==l

fp[0,m_/;m>0,k_/;k>0,l_/;l>0,0, 0,n_,o_/;o>0,p_, 1]:=
1/2^l/;p==Mod[l+1,B] && n==m && o==k

fp[0,m_/;m>0,k_/;k>0,l_/;l>0,0, 1,n_/;n>0,o_,p_,0]:=
(1-1/2^l) Po1[a,n-m] Po2[a,o-k+1] *
 1 /;p==l && n>=m && o>=k-1 && n<Q-1 && o<Q-1
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,0, 1,Q-1,o_,p_,0]:=
(1-1/2^l) (1-Sum[Po1[a,n],{n,0,Q-m-2}]) Po2[a,o-k+1] *
 1 /;p==l && o>=k-1 && o<Q-1
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,0, 1,n_/;n>0,Q-1,p_,0]:=
(1-1/2^l) (1-Sum[Po2[a,o],{o,0,Q-k-1}]) Po1[a,n-m] *
1 /;p==l && n>=m && n<Q-1
fp[0,m_/;m>0,k_/;k>0,l_/;l>0,0, 1,Q-1,Q-1,p_,0]:=
(1-1/2^l) (1-Sum[Po2[a,o],{o,0,Q-k-1}]) *
(1-Sum[Po1[a,n],{n,0,Q-m-2}])/;p==l

fp[0,m_/;m>0,k_/;k>0,0,l_/;l>0, 0,n_/;n>0,o_,1,p_]:=
1/2^l/;p==Mod[l+1,B] && n==m && o==k

fp[0,m_/;m>0,k_/;k>0,0,l_/;l>0, 1,n_,o_/;o>0,0,p_]:=
(1-1/2^l) Po1[a,n-m+1] Po2[a,o-k] *
 1 /;p==l && n>=m-1 && o>=k && n<Q-1 && o<Q-1
fp[0,m_/;m>0,k_/;k>0,0,l_/;l>0, 1,Q-1,o_/;o>0,0,p_]:=
(1-1/2^l) (1-Sum[Po1[a,n],{n,0,Q-m-1}]) Po2[a,o-k] *
 1 /;p==l && o>=k && o<Q-1
fp[0,m_/;m>0,k_/;k>0,0,l_/;l>0, 1,n_,Q-1,0,p_]:=
(1-1/2^l) (1-Sum[Po2[a,o],{o,0,Q-k-2}]) Po1[a,n-m+1] *
 1/;p==l && n>=m-1 && n<Q-1
fp[0,m_/;m>0,k_/;k>0,0,l_/;l>0, 1,Q-1,Q-1,0,p_]:=
(1-1/2^l) (1-Sum[Po2[a,o],{o,0,Q-k-2}]) *
(1-Sum[Po1[a,n],{n,0,Q-m-1}])/;p==l

fp[0,m_/;m>0,k_/;k>0,l_/;l>0,n_/;n>0, 0,o_,p_,r_,s_]:=
 Po1[a Ep[l,n],o-m] Po2[a Ep[l,n],p-k]*
```

```
 {i,1,Ix[1,Q-1,Q-1,B-1,B-1]+1}]
Find2[Lis_,index_]:=Table[If[Lis[[index,i]]!=0,Px[i-1],””],
 {i,1,Ix[1,Q-1,Q-1,B-1,B-1]+1}]
RD[b1_, b2_]:=2^Min[b1,b2]
p1[q1_,q2_]=2^(-Max[q1,q2])
p2[q1_,q2_]=Sum[2^(-q1-q2)*(2^q2-x),{x,1,RD[q1,q2]}]
totp[q1_,q2_]=p1[q1,q2]+p2[q1,q2]+p2[q2,q1]
Ep1[q1_,q2_]=(RD[q1,q2]-1)/2
Ep2[q1_,q2_]:=2^(-q1-q2) Sum[i (2^q2-i),{i,1,RD[q1,q2]}]
Ep[q1_,q2_]=Ep1[q1,q2]+Ep2[q1,q2]+Ep2[q2,q1]
(*
Eep1[q1_,q2_,n_]:=Sum[Po1[i a,n],{i,0,RD[q1,q2]-1}] 2^Max[-q1,-q2]
Eep2[q1_,q2_,n_]:=2^(-q1-q2) Sum[Po1[i a,n] (2^q2-i),{i,1,RD[q1,q2]}]
Eep[q1_,q2_,n_]:=Eep1[q1,q2,n]+Eep2[q1,q2,n]+Eep2[q2,q1,n]
Ptb=Table[Table[Eep[q1,q2,i]/Po1[Ep[q1,q2]a,i],{i,1,10}],{q1,1,10},{q2,1,10}]
*)
(* Transition probability function *)
fp[0,0,0,0,0, 1,0,0,0,0]=1-(1-Po1[a,0])(1- Po2[a,0])
fp[0,0,0,0,0, 0,k_/;k>0,l_/;l>0,0,0]=Po1[a,k] Po2[a,l]

fp[1,0,0,0,0, 0,0,0,0,0]=Po1[1,0] Po2[1,0]
fp[1,0,0,0,0, 1,0,0,0,0]=Po1[1,1] Po2[1,0]+Po1[1,0] Po2[1,1]
fp[1,0,0,0,0, 1,k_/;k>0,0,0,0]=Po1[1,k+1]Po2[1,0]
fp[1,0,0,0,0, 1,0,k_/;k>0,0,0]=Po1[1,0]Po2[1,k+1]
fp[1,0,0,0,0, 0,k_/;k>0, l_/;l>0,0,0]=Po1[1,k]Po2[1,l]

fp[0,m_/;m>0,k_/;k>0,0,0, 0,l_/;l>0,n_/;n>0,1,1]:=Po1[a,l-m] *
 Po2[a,n-k] /; l>=m && n>=k &&l<Q-1&&n<Q-1
fp[0,m_/;m>0,k_/;k>0,0,0, 0,Q-1,n_/;n>0,1,1]:=Po2[a,n-k] *
(1- Sum[Po1[a,l],{l,0,Q-m-2}])/; n>=k && n<Q-1
fp[0,m_/;m>0,k_/;k>0,0,0, 0,l_/;l>0,Q-1,1,1]:=Po1[a,l-m] *
(1- Sum[Po2[a,n],{n,0,Q-k-2}])/; l>=m && l<Q-1
fp[0,m_/;m>0,k_/;k>0,0,0, 0,Q-1,Q-1,1,1]:=
 (1-Sum[Po1[a,l],{l,0,Q-m-2}]) *
 (1- Sum[Po2[a,n],{n,0,Q-k-2}])

fp[0,m_/;m>0,0,l_/;l>0,0, 0,n_,k_/;k>0,o_,1]:=Po1[a 2^(l-1),n-m] *
 1/2^l Po2[a,k]/;n>=m && o==Mod[l+1,B]
fp[0,m_/;m>0,0,l_/;l>0,0, 1,n_,0,0, 0]:=Po2[2^(l-1) a,0] *
Po1[2^(l-1) a, n-m+1]/;n>=m-1 && n<Q-1
fp[0,m_/;m>0,0,l_/;l>0,0, 1,Q-1,0,0, 0]:=Po2[2^(l-1) a,0] *
(1-Sum[Po1[2^(l-1) a, n],{n,0,Q-m-1}])
fp[0,m_/;m>0,0,l_/;l>0,0, 1,n_,o_,p_, 0]:=Po2[2^(l-1) a,o+1] *
Po1[2^(l-1) a, n-m]/; n>=m && p==l && n<Q-1
fp[0,m_/;m>0,0,l_/;l>0,0, 1,Q-1,o_,p_, 0]:=Po2[2^(l-1) a,o+1] *
(1-Sum[Po1[2^(l-1) a, n], {n,0,Q-m-2}])/; p==l
```

Appendix A: Program for a Two-Station Model

```
a=1/20
l1=0.2
l2=0.2
Q=3
B=3
(* The Index mapping functions*)
(*The following constraints must hold:
 0<= t < 2
 0<=q1,q2<Q
 0<=b1,b2<B *)
Ix[t_, q1_, q2_, b1_, b2_]=t B^2 Q^2 + (q1 Q + q2)B^2 + b1 B + b2
tp[indx_]=Quotient[indx , Q^2 B^2]
q1[indx_]=Quotient[Mod[indx, Q^2 B^2], Q B^2]
q2[indx_]=Quotient[Mod[indx, Q B^2], B^2]
b1[indx_]=Quotient[Mod[indx, B^2 ], B]
b2[indx_]=Mod[indx, B]
Px[N_]={tp[N], q1[N], q2[N], b1[N], b2[N]}
Pois1[t_, n_/;n>=0]:=(l1 t)^n/n! Exp[-l1 t]
Pois2[t_, n_/;n>=0]:=(l2 t)^n/n! Exp[-l2 t]
Po1=Compile[{t,n}, Pois1[t,n]]
Po2=Compile[{t,n}, Pois2[t,n]]
Posit[Lis_,x_,i_]:=i-1/;Abs[First[Rest[Lis[[i]]]]-x]<0.0001
Posit[Lis_,x_,i_]:=””
Pos[Lis_,x_]:=Table[Posit[Lis,x,i],{i,1,Ix[1,Q-1,Q-1,B-1,B-1]+1}]
Loc[Lis_,x_,i_]:=Px[i-1]/;Lis[[i]]==x
Loc[Lis_,x_,i_]:=””
Location[Lis_,x_]:=Table[Loc[Lis,x,i],{i,1,Length[Lis]}]
Location::usage=
 “Location[Lis_,x_] finds the location of the SYMBOL x in the
 list Lis”
Rcut::usage=
 “Rcut[x,e] rounds the number less than e to 0”
Rcut[x_,e_]:=If[x>e,x,0]
Pos0[Lis_,i_]:=Px[i-1] /;N[First[Rest[Lis[[i]]]],5]==0.00
Pos0[Lis_,i_]:=””
NullSpc:=Table[Pos0[d,i],{i,1,Ix[1,Q-1,Q-1,B-1,B-1]+1}]
Count0[Lis_]:=Count[Lis,0.]
ZeroSize[Lis_]:=Apply[Plus,Map[Count0,Lis]]
FindNonZero::usage =
 “FindNonZero[list,index] finds the position of the non-zero transition
 to state pointed by index; index can be formed from the coefficients
 of the state components by adding one to the function Ix”
FindNonZero[Lis_,index_]:=Table[If[Lis[[i,index]]!=0,Px[i-1],””],
```

References

[1] G. T. Almes, E. D. Lazowska, *The Behavior of the Ethernet-like Computer Communication Networks,* Proceedings of the Local Area Communications Network Symposium, pages 66-81. IEEE, 1979.

[2] R. M. Metcalfe, D. R. Boggs, *Ethernet: Distributed Packet Switching for Local Computer Networks,* Communications of the ACM, Vol. **19** No. 7, pp 395-404, July, 1976.

[3] E. D. Lazowska, J. Zahorian, G. S. Graham, K. C. Sevcik, *Quantitative System Performance: Computer System Analysis Using Queueing Network Models,* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1985.

[4] K. Sohraby, M.L. Molle, A. N. Venetsanopoulus, *Comments on "Throughput Analysis for Persistent CSMA Systems",* IEEE Trans. Communications, Vol. **COM-35** No. 2, pp 241-243, February, 1987.

[5] S. Armyros. *On the Behavior of Ethernet: Are Existing Analytic Models Adequate?.* Master's thesis, University of Toronto, Department of Computer Science, 1991.

[6] H. Takagi, L. Kleinrock, *Throughput Analysis for Persistent CSMA Systems,* IEEE Trans. Communications, Vol. **COM-33** No. 7, pp 627-638, July, 1985.

[7] H. Takagi, L. Kleinrock, *Output Processes in Contention Based Packet Broadcast Systems,* IEEE Trans. Communications, Vol. **COM-33** No. 11, pp 1191-1199, November, 1985.

[8] H. Takagi, L. Kleinrock, *Mean Packet Queueing Delay in a Buffered Two-user CSMA/CD System,* IEEE Trans. Communications, Vol. **COM-33** No. 10, pp 1136-1139, October, 1985.

[9] H. Takagi, L. Kleinrock, *Diffusion Process Approximation for the Queueing Delay in Contention Packet Broadcasting Systems,* Performance of Computer-Communication Systems, pages 111-124 . North-Holland, 1984.

3.7 Conclusions

The model for describing two stations competing for a channel under a CSMA/CD policy and the Exponential Backoff is elaborated in this chapter. It is a embedded Markov chain model, for which the state transitions take a place only when the station starts transmitting a packet. On the network, it can result with a success or a collision and the state transition will depend on the outcome of this event.

The state space is formed to describe the state of the network (transmission or not), the queues and backoff counters, in both of the stations. The large state space generated by these parameters is reduced to a simplest possible, by finding the set of impossible states and excluding the corresponding rows and columns from the Markov matrix.

Two major approximations has been made: the increase in the population is based on the average length of the period, instead of using the more sophisticated (and computationally more costly) calculations, and the residual time of the station in the baklogged state is neglected.

The model presented requires a lot of computing time. Solving the system given by one pair of input rates and the smallest queues and backoff counters (three) takes about twenty minutes on the Sparc2 workstation. When going to the more realistic models, the state space increases dramatically. The dimensionality of Markov matrix is in order of 20000 states, in a full (buffer capacity 16, backoff counter 16) configuration. This is the reason why we believe that the massively parallel computer will be necessary for dealing with this type of models in a future.

FIGURE 16 Throughput as a Function of Load

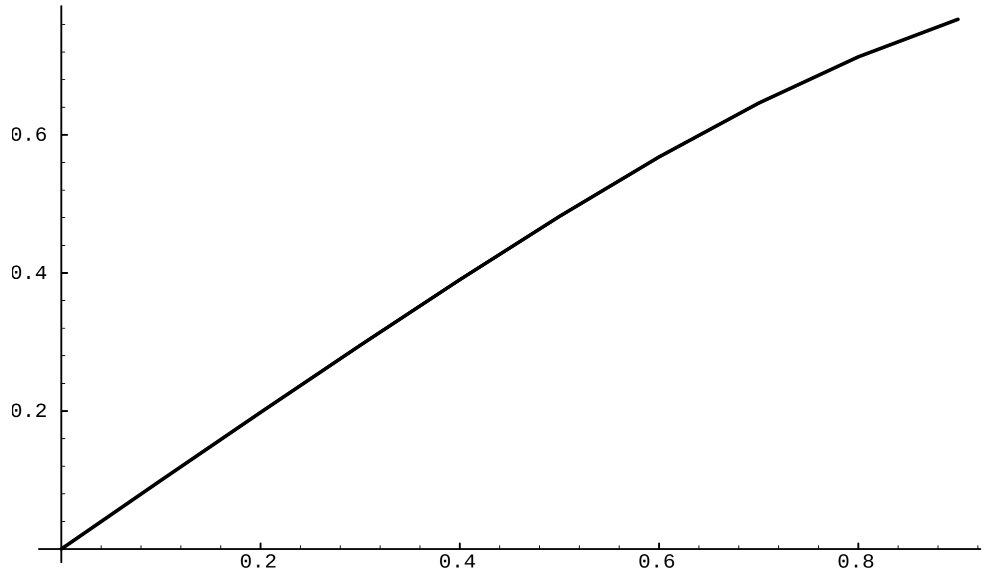
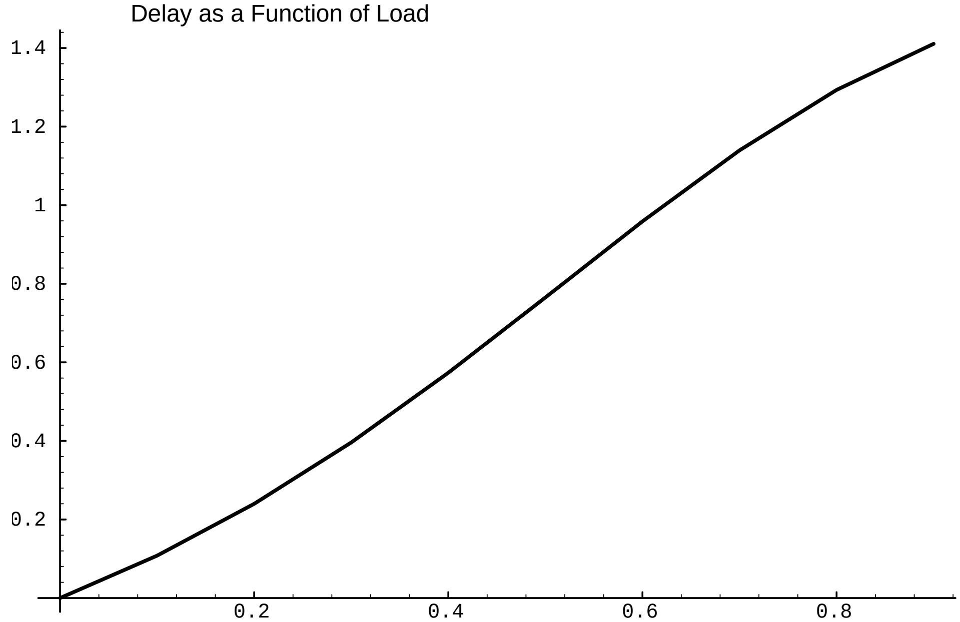


FIGURE 17 Delay as a Function of Load



Here, the throughput function has the expected shape, while the delay characteristics tends to underestimate the delay under high load. The reason is in very small queue size (2). After some load, most of the packets will be discarded and they will not contribute to the mean queue length. With deeper buffers, more realistic results will be obtained.

Figure 14 and Figure 15 show the state distribution in case of heavy load- 0. 8 of the total network capacity.

FIGURE 14 Sate Distribution- Load 0.8- all the States

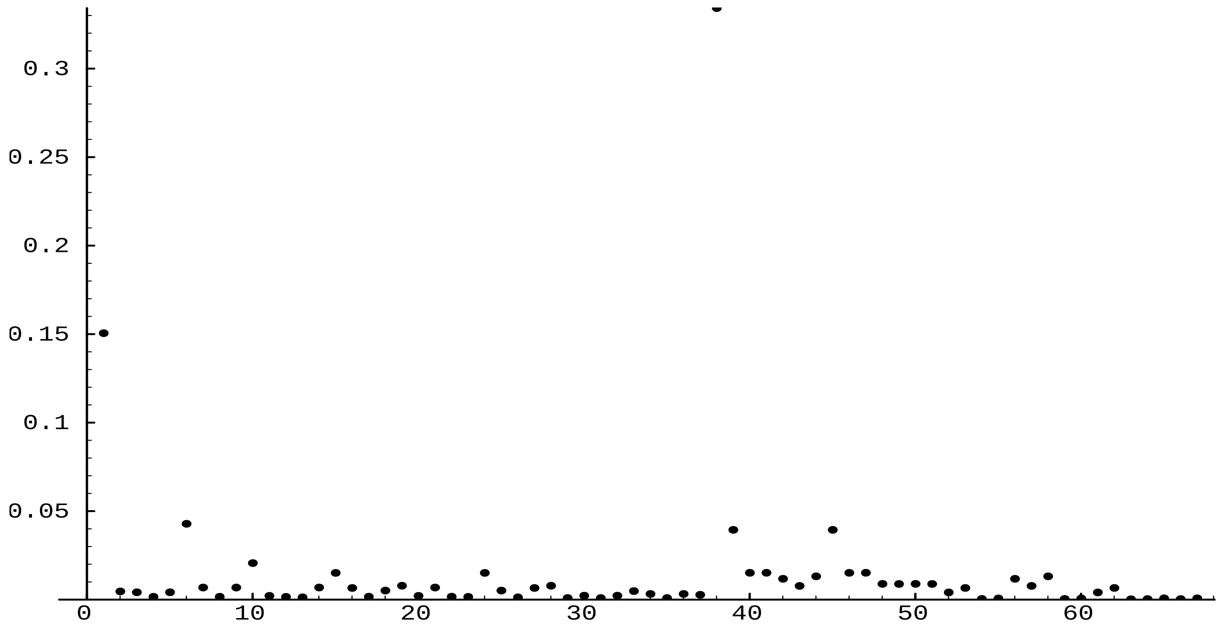
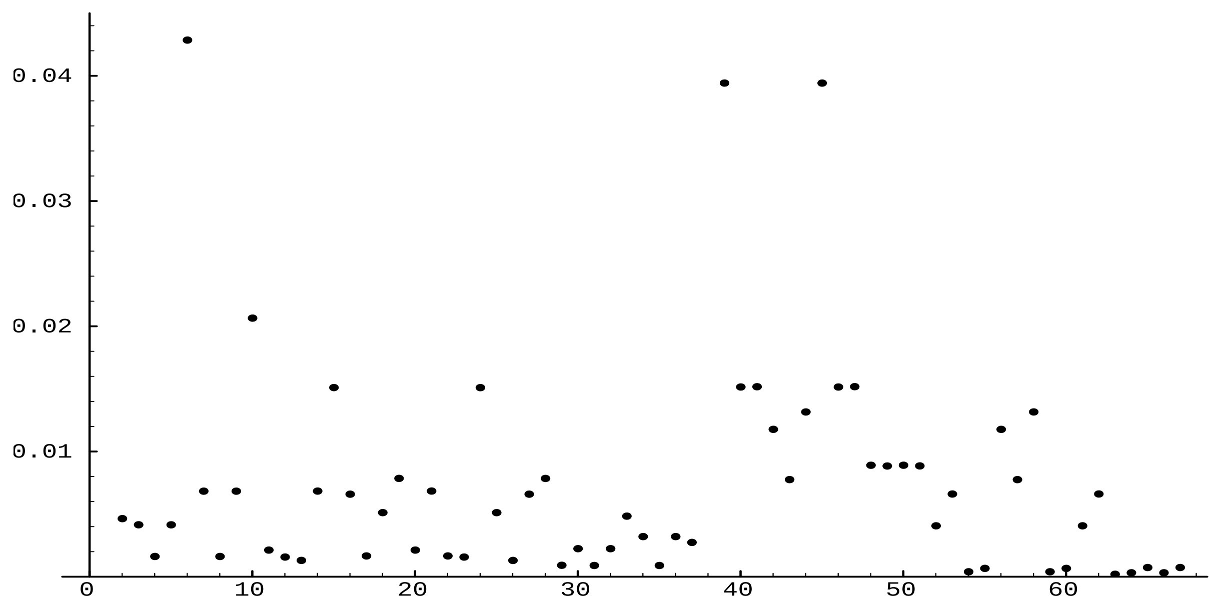
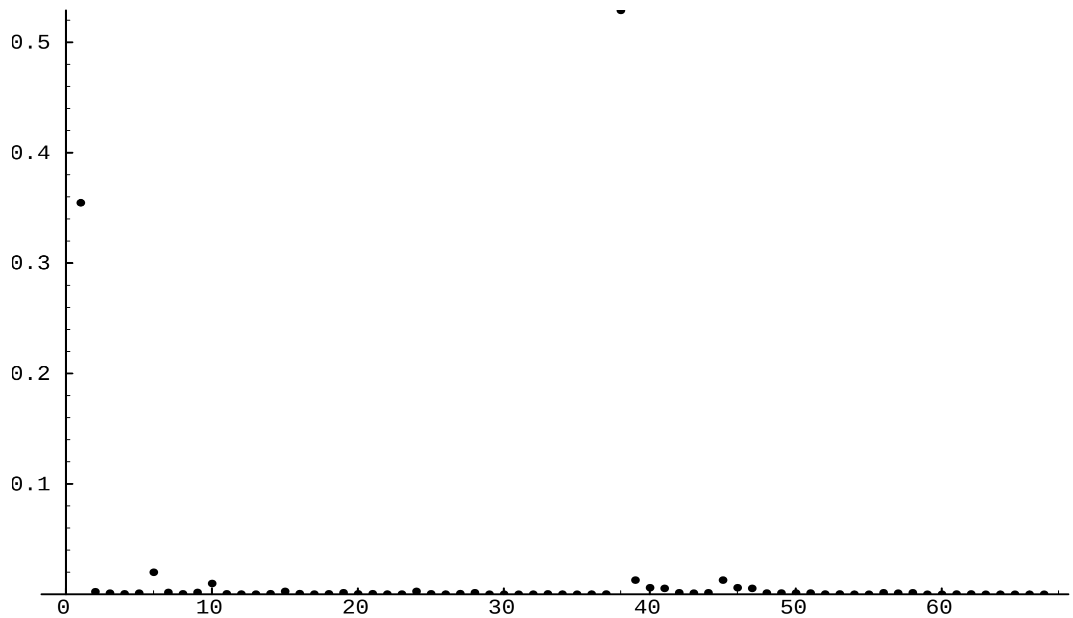


FIGURE 15 State Distribution- Load 0.8- Truncated Graph



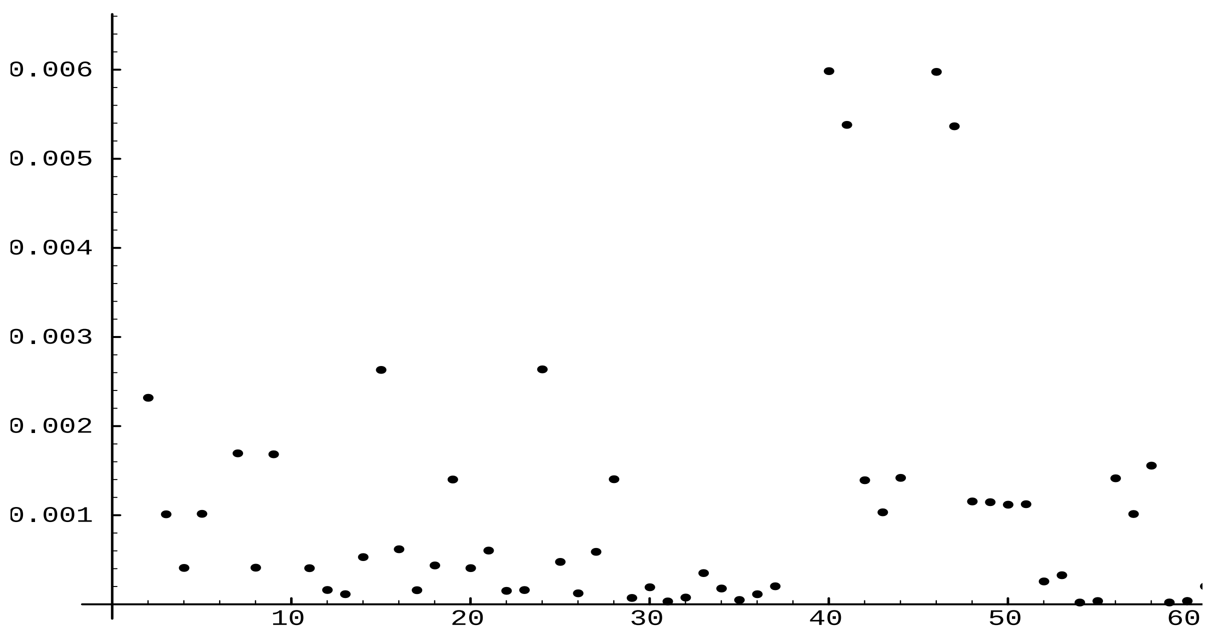
Knowing the state distribution, we can calculate throughput and the average waiting time, as discussed before. The curves are given in next two Figures.

FIGURE 12 State Space Distribution- Load 0.4- All the States



The complete state distribution is shown in Figure 12 , in a case of total load of 0.4. The states are encoded, but the state 0 is the most occupied state. Figure 13 shows the same situation, with the most probable states truncated.

FIGURE 13 State Distribution- Load 0.4- Truncated Graph



$$\beta(0,0,0,0,0) = \frac{1}{l1 + l2} \quad \text{(EQ 20)}$$

Here, the last value is the expected idle time in the network, while the former two equations describe the durations of the successful subcycle and the collision resolution subcycle, respectively.

The states will contribute to the success by the amount

$U(i), i \in \{1, N\}$

Defined by the help of the following auxiliary function:

$$\Upsilon(1, q1, q2, b1, b2) = 1 \quad \text{(EQ 21)}$$

$$\Upsilon(0, q1, q2, b1, b2) = 0 \quad \text{(EQ 22)}$$

When we find the solution of the system (the equilibrium point probabilities)

$p_i, i \in \{1, N\}$,

the only thing remaining is to calculate the expected successful time and to divide it by the expected duration of the cycle, i.e.

$$Through = \frac{E(U)}{E(B)} = \frac{\sum U(i) \times p_i}{\sum B(i) \times p_i} \quad \text{(EQ 23)}$$

Here, the summation is performed over all of the states. This model allow us to calculate the delay characteristics, too. The Little's formula will be used, so that the number of the packets in queues will be divided by number of the packets which exit the system. With every state, the value Q(i) will denote the number of the packets held in queue, while the X(i) will be the number of the packets which have been successfully transmitted being in that state. Then the formula for the mean delay is given by:

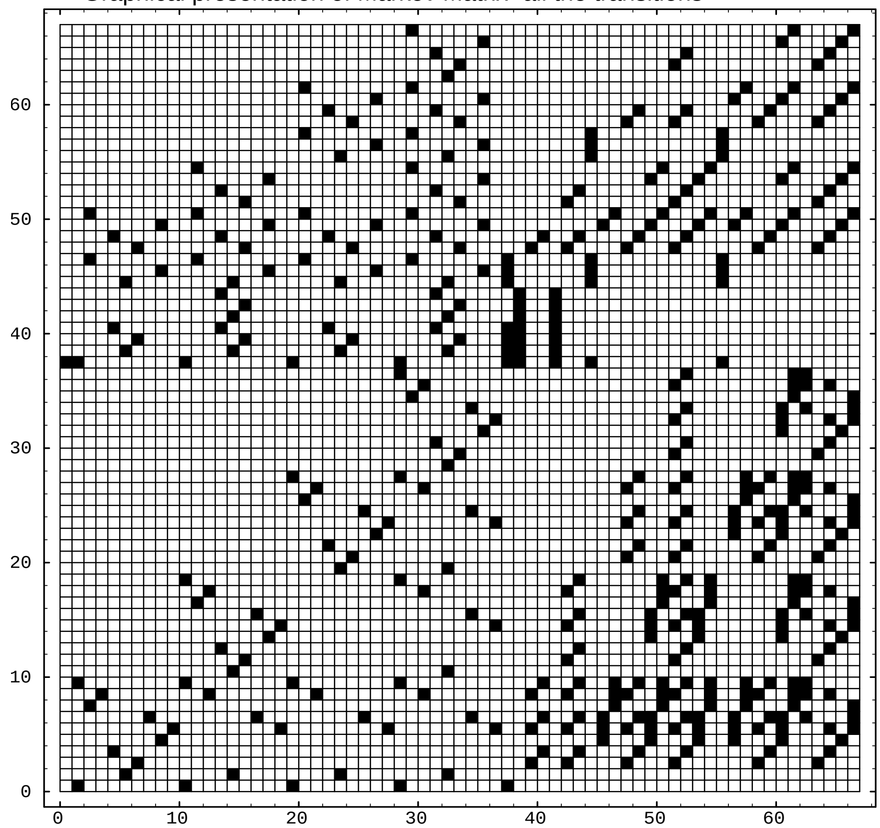
$$Delay = \frac{E(Q)}{E(X)} = \frac{\sum Q(i) \times p_i}{\sum X(i) \times p_i} \quad \text{(EQ 24)}$$

3.6 The Results

Having the model solved we can discuss and analyze the results. Figure 10 and Figure 11 contains the graphical representation of the state transition matrix. It is important to notice that the state space is "compressed", i.e. the impossible states are excluded from the state space. From them, it is possible to get some idea about the solution.

The results for two loads are given next. The main difference in the stationary probabilities can be seen when considering the state *(0,0,0,0,0)* and *(1,0,0,0,0)*. In the graphs, they are encoded as the state 0 and 38.

FIGURE 11 Graphical presentation of Markov Matrix- all the transitions



3.5 The Solution

Having defined the transition matrix, we can find the stationary probabilities. Let N be a finite number of states in the system. The durations of them are, in general, are the random variables. We will denote the expected values by

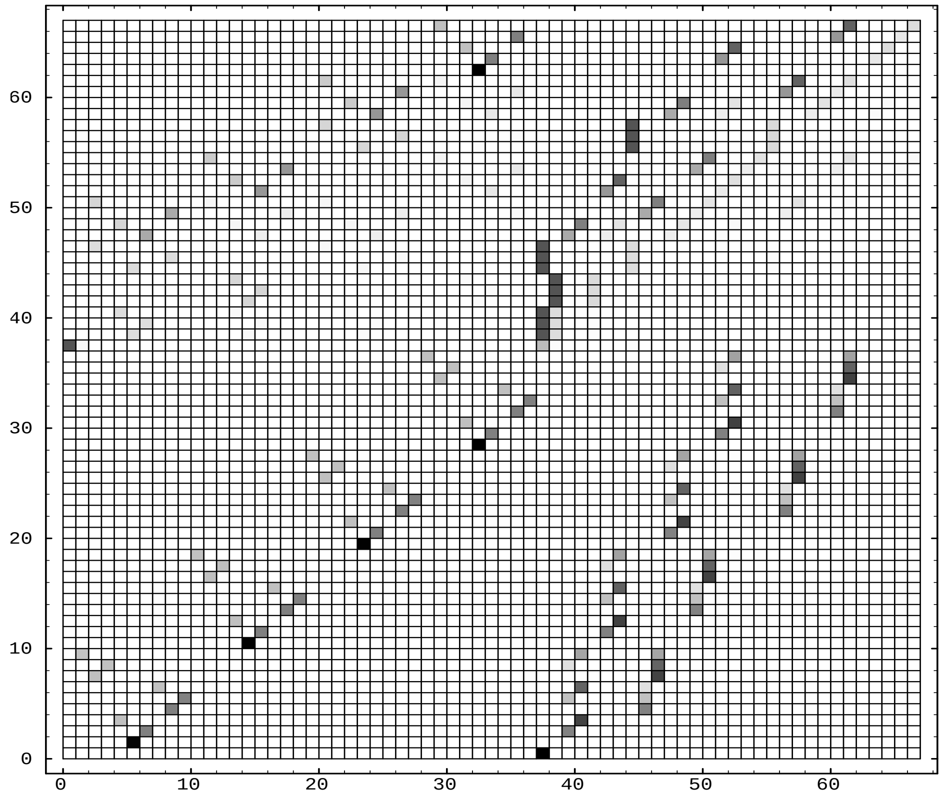
$$B(i), i \in \{1, N\}$$

The value B(i) is defined by the following scheme. Instead of labelling states with a number, we will use the 5-tuple notation and new function β defined over the set of 5-tuples. Obviously, there is an one-to-one correspondence between these two notations. Here is how this function is defined

$$\beta(1, q1, q2, b1, b2) = 1 + a \quad \text{(EQ 18)}$$

$$\beta(0, q1, q2, b1, b2) = a \times 2^{Min(b1, b2)} \quad \text{(EQ 19)}$$

FIGURE 10 Graphical presentation of Markov matrix- the intensities



In case that any of the numbers on the right-hand size is equal to the Q-1, the approach described in Equation 21 will be used. From the same state, it is possible to reach the state (1,q1,0,0,0) and (1,0,q2,0,0). The way to describe both of the cases will be similar, as in following formula:

$$P\{ (1, 0, 0, 0, 0) \to (1, q1,0,0,0) \} = P_1(1, q1) \times P_2(1, 0) ,q1 < Q-1$$

The equation says that the success will follow the successful transmission if there are new arrivals at exactly one station.

In all the other cases, the determination of the transition probabilities will be straightforward. There is one more approximation used, by which we neglect the residual lifetime of the station when the another one is transmitting a packet. We could avoid this approximation, but the state space will increase dramatically. The complete listing is contained in source code written in Mathematica which is given in Appendix A.

Figure 10 shows graphically the transition matrix, in a case Q=3, B=3. The intensities are denoted by the various levels of gray colour. After that, all the places in the matrix which are nonzero are coloured black, in Figure 11 .

The state space is “compressed”, i.e. all the impossible states are not listed.

3.4 Transition Rates

In order to simplify the description of the transition rates, we will describe only the few common characteristics first and then illustrate it by two characteristic examples. We are aware that the complete description will be lengthy and not too insightful. However, the Appendix A contains the complete code written in Mathematica, and it can serve as a reference.

Generally, the transition probabilities will depend on the difference between the number of packets in queue at the beginning and the end of the observed period, i.e.:

$$Parr(t, \Delta i) = \frac{(lt)^{\Delta i} e^{-lt}}{\Delta i!} \qquad \textbf{(EQ 16)}$$

where t is the expected duration of that cycle, and $P_{arr}(t,i)$ is the probability that i packets will arrive in that period. By using the expected duration of the cycle, we are making the approximation, which can be very rough one. The increase in the population

One unusual situation is the buffer overflow. In all of the transitions, there is a chance that the number of the packets which arrive between the transitions will exceed the remaining free space in the buffer. If there were q packets in a buffer, then the probability that the buffer will be full is given by the following:

$$P_{full}(t, \Delta i) = 1 - \sum_{i=0}^{Q-q-2} P_{arr}(t, \Delta i) \qquad \textbf{(EQ 17)}$$

It is obvoius that in a case of successful departure, the difference between the number of packets in starting and finishing state will be increased by one, before using any of the two previous formulas.

In order to simplify the notation, we will introduce functions P_1 and P_2. They are defined in a same way as P_{arr}, but the load factor l will be replaced by factors $l1$ and $l2$, respectively. Few characteristic examples will illustrate how the Markov matrix of the system is formed.

Since the arrival processes in both queues are independent, the transition probabilities between two states will be given as a product of transition probabilities for every station. Following example shows how these probabilities are calculated.

Starting from the state (1,0,0,0,0), the transition probability to the state (0,q1,q1,0,0) will be:

$$P\{(1,0,0,0,0) \rightarrow (0, q1, q2, 0, 0)\} = P_1(1, q1) \times P_2(1, q2), q1, q2 < Q-1$$

The equation says that the probability to reach a state (0,q1,q2,0,0) is equal to the product of probability of q1 arrivals in the first queue and the probability of q2 arrivals in the second one during the ongoing transmission. The duration of this state is *1*.

TABLE 2 Transition Possibilities

State Group	Next Possible State Group
(1,q1,0,0,0)	(1,q1,0,0,0), (0,q1,q2,1,1)
(1,0,q2,0,0)	(1,0,q2,0,0), (0,q1,q2,1,1)
(1,q1,q2,0,0)	(0,q1,q2,1,1)
(1,q1,0,b1,0)	(0,q1,q2,b1,1), (1,q1,0,0,0)
(1,0,q2,0,b2)	(0,q1,q2,1,b2), (1,0,q2,0,0)
(1,q1,q2,b1,0)	(1,q1,q2,b1,0), (0,q1,q2,b1,1)
(1,q1,q2,0,b2)	(1,q1,q2,0,b2), (0,q1,q2,1,b2)

Here, *q1*,*q2*,*b1* and *b2* denote that any number will take a place. It means that *q1* in the left column and *q1* in the right column do not represent the same number, but any non-zero value.

Since the state space is huge, we are not able to show all the states and transitions. We are including one example state diagram, showing the subset of states and transitions just to illustrate how the state machine is developing through the time.

FIGURE 9 Subset of State Machine Diagram

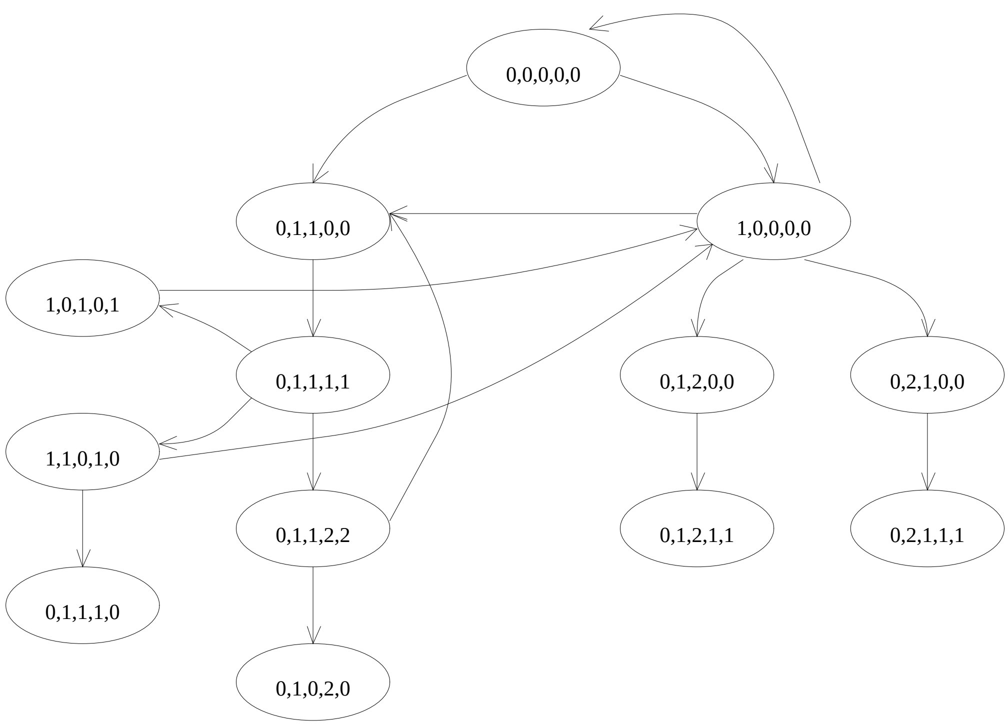


TABLE 1 State Space Groups

State Group	Description
(0,0,0,0,0)	The zero state-all the queues and counters are empty
(1,0,0,0,0)	Transmission in progress - the queues and counters are empty
(0,q1,q2,0,0)	Two station ready to trasmit, no previous collisions
(0,q1,0,b1,0)	Station 1 has packets delayed due to previous collisions
(0,0,q2,0,b2)	Station 2 has packets delayed due to previous collisions
(0,q1,q2,b1,0)	Both stations have packets to send, station 1 has to wait
(0,q1,q2,0,b2)	Both stations have packets to send, station2 has to wait
(0,q1,q2,b1,b2)	Both stations have packets to send, both have to wait
(1,q1,0,0,0)	Transmission- station 1 has nonempty buffer
(1,0,q2,0,0)	Transmission- station 2 has nonempty buffer
(1,q1,q2,0,0)	Transmission, both stations have non-empty queues,
(1,q1,0,b1,0)	Transmission, station 1 has to wait
(1,0,q2,0,b2)	Transmission, station2 has to wait
(1,q1,q2,b1,0)	Transmission, station 2 will transmit next
(1,q1,q2,0,b2)	Transmission, station1 will transmit next

3.3.2 Transition Description

In order to avoid confusion in describing the transition probabilities, we will describe some common characteristics first. First, all the groups are handled in an uniform manner. From one group representative, the transition will take place only to a certain number of state groups. The following Table shows the way in which the group representative choses the next state group

TABLE 2 Transition Possibilities

State Group	Next Possible State Group
(0,0,0,0,0)	(1,0,0,0,0), (0,q1,q2,0,0)
(1,0,0,0,0)	(0,0,0,0,0), (1,0,0,0,0), (1,q1,0,0,0), (1,0,q2,0,0), (0,q1,q2,0,0)
(0,q1,q2,0,0)	(0,q1,q2,1,1),
(0,q1,0,b1,0)	(0,q1,q2,b1,1), (1,q1,0,0,0), (1,q1,q2,b1,0)
(0,0,q2,0,b2)	(0,q1,q2,1,b2), (1,0,q2,0,0), (1,q1,q2,0,b2)
(0,q1,q2,b1,0)	(0,q1,q2,b1,1), (1, q1, q2, b1,0),
(0,q1,q2,0,b2)	(0,q1,q2,1,b2), (1,q1,q2,0,b2)
(0,q1,q2,b1,b2)	(0,q1,q2,b1,b2), (1,q1,q2,b1, 0), (1,q1,q2,0,b2)

the events on the network will take a place only after integer number of slots, which are usually defined to be *2a* units long. Here, *a* is the channel propagation delay. The packet transmission will last *1* units of time.

3.3 The State Space Organization and the Transition Probabilities

The configuration from Figure 8 can be modelled as an embedded Markov chain. The instants of the departure or the collision will be considered as the moments at which the transitions occur. In order to represent the information on the network status, the following state space vectors will be used.

$$\chi = (t, q1, q2, b1, b2) \qquad \text{(EQ 15)}$$

Here:

$t \in \{0, 1\}$

$q1, q2 \in \{0, 1, \ldots Q-1\}$

$b1, b2 \in \{0, 1, \ldots B-1\}$

The first value will be set to one if one of the stations is transmitting a packet. Otherwise, (the silent channel, or the collision) it will be zero. The queue content and the backoff counter description is given by the numbers *bi* and *qi* at the station i.

It is important to notice that not all the states are possible. For example, it is not possible to have *bi* greater than zero if the queue *i* is empty. This will help us to reduce the state space considerately. Nevertheless, here is a huge state space defined by the 5-tuple given above.

In order to define all the transition probabilities, we will structure the state space into few classes. The transition between the elements of the same class will be defined in a same way.

3.3.1 State Space Organization

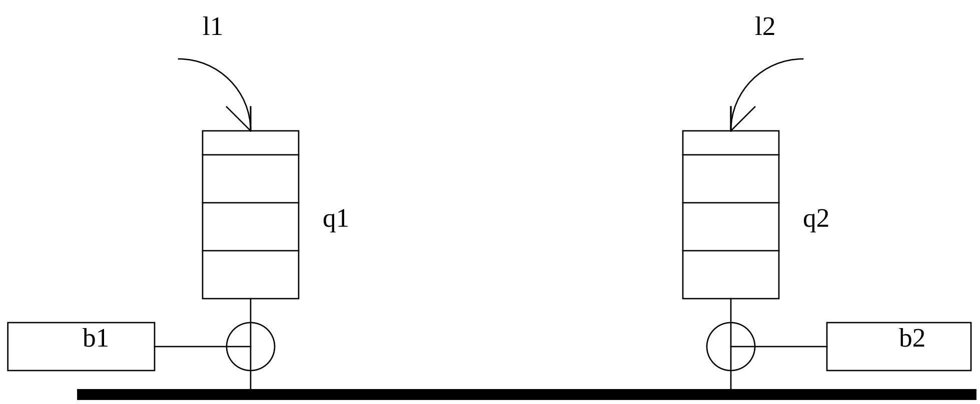
The following Table shows the state space groups. All the groups have the uniform behaviour, i.e. the transition probabilities to the groups of states are written in an uniform manner. All the parameters used (*q1,q2,b1,b2*) are nonzero, when mentioned.

The information about the originator of the message is generally not contained in the 5-tuples. In some cases, we can determine which station is transmitting. In state group characterized by the state vector (1,q1,q2,0,b2) for example, we can be sure that the station 1 is transmitting, since the backoff counter is always setup to zero at the station transmitting, and it this case it is obvious that the station 2 (backoff counter contents is greater than zero) does not transmit a packet.

3.2 The Model

In order to model the interaction between the stations, we will take a look at the details of the protocol operation, when of two stations are attached to a network. Figure 8 shows the configuration we intend to model.

FIGURE 8 Model- Configuration



There are two stations attached to a network. Their state is characterized with number of packets in a queue and the backoff counter contents. They are denoted by *q1*, *q2*, *b1* and *b2*, respectively. The input to the stations is modelled by Poisson processes, with parameters *l1* and *l2*.

The operation of the system is defined as in standard Ethernet. If the backoff counter is equal to zero, the station will try to send the packets held in its queue. If there is a collision during this attempt, the backoff counter will be set to one, and the station will go to the *backlogged* state in which it will wait for a random amount of time before new retransmission attempt. We will deal with the exponential policy, which means that the station *i*, with the backoff counter's content equal to *bi* will wait up to 2^{bi} *slot times*. With every new collision, the counter contents will be increased.After the station successfully transmits the packet, the counter is reset to zero, and the station goes to the nonbacklogged state.

Let the buffer size be *Q*, and the maximal content of the backoff counter *B*. It is possible to have the *buffer overflow*, if too many packets arrive in a short period of time. Also note that packets are lost if they suffer too many collisions in succession. Thus, the backoff counter will be *reset to zero* when the maximum value is reached. The model described here will be the *slotted model*. Starting from every collision on the network,

CHAPTER 3 Two-Station Model

This chapter describes a detailed Markov model for two stations connected to the Ethernet. The model is the improvement over existing ones because it models closely the operation of the backoff algorithm. We will deal exclusively with the binary exponential backoff, but all the other policies can be incorporated. The motivation for this model and the previous works are reviewed in first part, followed by detailed description of the model. A set of the results closes this chapter.

3.1 Introduction

In Chapter 3, we pointed out that no previous model of CSMA/CD is capable of describing the interactions between the stations using the exponential backoff policy. It means that we are not able to describe the queue interactions and the blocking of the channel by one station, the effects which very often dominate the traffic characteristics of the Ethernet. That is the reason to try to make the more realistic model. We will consider the configuration which consists of only two stations, and try to model all the details as accurately as possible.

There have been few attempts in the past to obtain a realistic CSMA model, and the conclusion was [8] that the queue interactions allow only two-station models to be viable. The authors produced a transform-based solution for the joint queue lengths in that case. The model developed can not be extended to more than two stations in a network, even though this model does not take the backoff algorithm into account.

Most of the previous models had the underlying assumption of symmetrical load incorporated in it. We tried to allow any pattern of the traffic to be modelled. In our model, the individual loads can be setup independently, but we still assume it is describable as a Poisson process.

2.4 Conclusions

This chapter offers two ways to deal with the effects which are intrinsic to the backoff algorithm. Both of them are based on the simple adaptation of the existing scheme.

2.4.1 Extensions to the Infinite Population Model

In first case, the cycle in which the events are observed are redefined from observing the network events to the events as seen by the station. It means that the time difference between two attempts to transmit is not the expected time of the idle cycle on the network, but the retry delay, according to the backoff protocol. This approach can also be used when considering various models proposed by Takagi and Kleinrock [6](which is the “imperfect” precedent to Sohraby’s work) and [7],[8]. Our opinion is that the framework established in Takagi’s work [9] can easily incorporate approach as used here. Takagi’s idea is to decompose the system composed of buffered stations and the network in a way that only the mean value and the second moment of the interdepature times from the network are used in calculating the queue lengths. They are calculated by the diffusion approximation method. This model can be valid in description of asymmetrical users.

Second extension of the model can be in building the finite population model. In doing this, the interaction of users can not be taken into account. That is the reason for which we guess that this model will not be too useful.

2.4.2 Extensions to the Markov Chain Approximate Model

When trying to make the model proposed in Section 2.3 on page 13 usable in wider class of cases, model losses its simplicity. We will discuss the attempts to extend the same approach to the finite population and the asymmetric load cases.

The fact we have to model is the decreased “primary” load on the network when the station transmits or competes for the channel. In the asymmetric load case the amount for which the load is to be decreased is sorted in decreasing way, since the station with the greater load will start transmitting earlier, in the average.

However, in all these extension the issue of modelling the queue interactions and effects which arise from coupling the stations in retries stays unresolved. This is the reason to try to make more realistic model. Next section describes one such model.

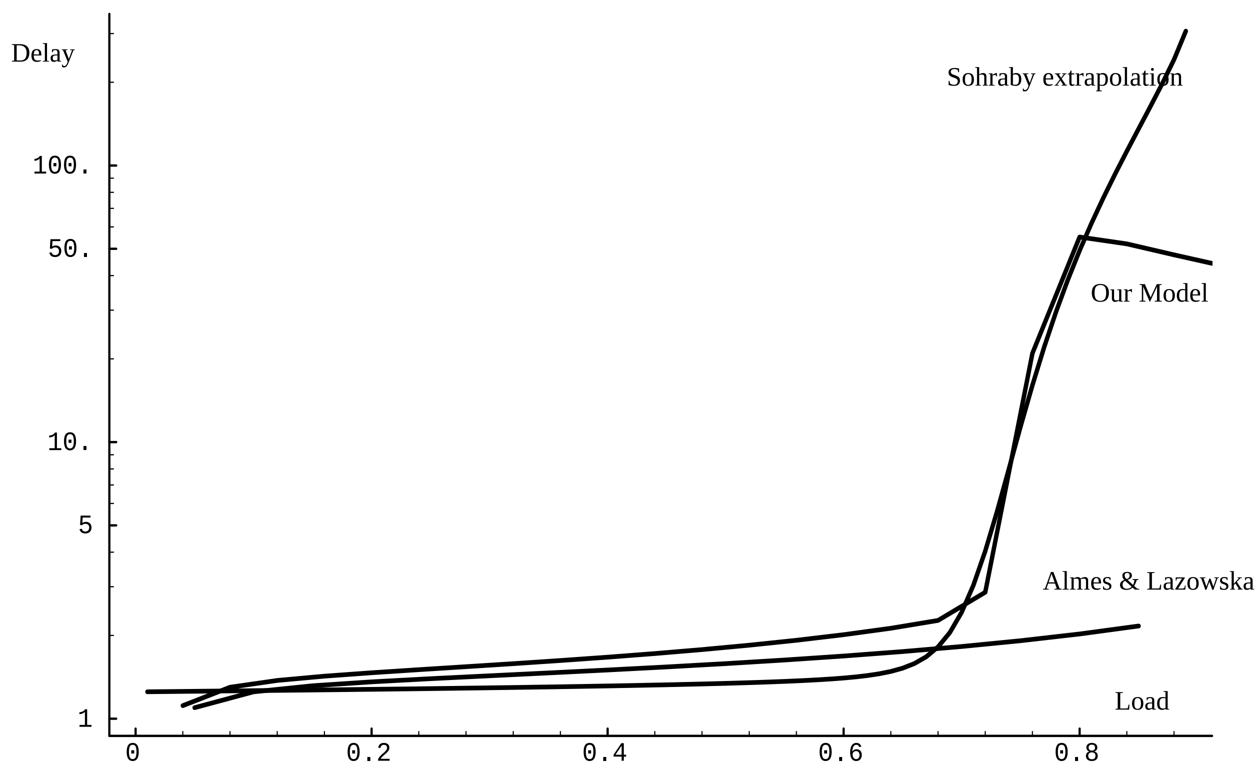
Having the solution of the system available, we can, using the Little's formula, calculate the delay in the system. It is given by following formula:

$$D = \frac{\sum_{q \in Q} P(q) \bullet q}{load} \quad \text{(EQ 14)}$$

Where the expected number of packets in the system is divided by its load.

Next Figure provide us the comparison between three methods described here. Our first approximate model and our Markov chain model show close characteristics, while the Markov chain model proposed by Almes and Lazowska show significant mismatching with both of these models. The decrease in delay in our model is caused by the fact that we are able to model only finite number of states in the system. With increasing this number, the model is becoming closer to the extrapolated Sohraby's model, but the computing time also increases dramatically.

FIGURE 7 Comparison of three models



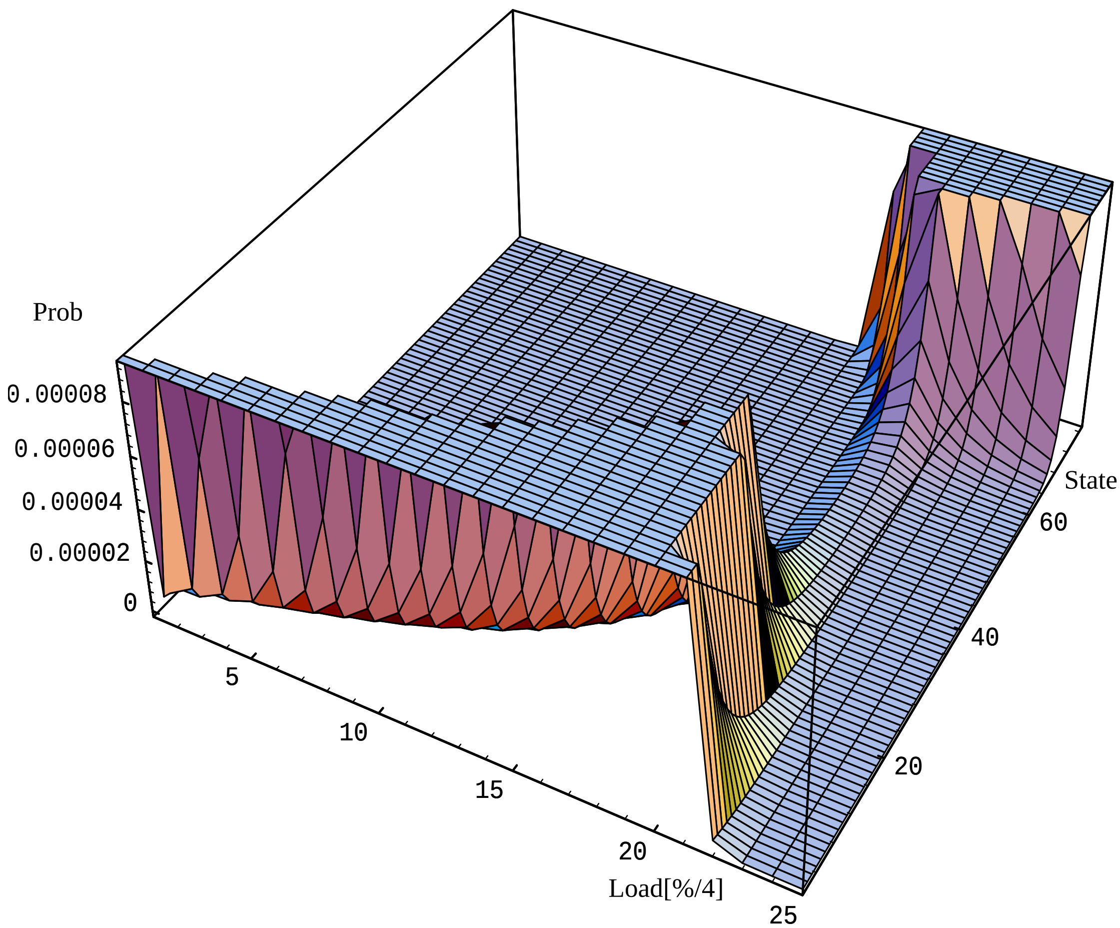
$$p(k) = \frac{load^k}{\prod_{i \in states} through(i) \times p0(k)} \qquad \textbf{(EQ 12)}$$

Where p0 is given by:

$$p0(k) = \frac{1}{1 + \sum_{l \in states} \frac{load^l}{\prod_{i=1}^{l-1} through(i+1)}} \qquad \textbf{(EQ 13)}$$

Next Figure shows how the state probability is distributed, as a function of a network load, in a 70-states model.

FIGURE 6 State Probability Distribution as a Function of a Load (in percents/4)



$$m(n) = \frac{1}{1 + W(n) \bullet 2a} \quad \text{(EQ 9)}$$

Where *l* is the packet generation rate, and the *m(n)* is defined as the conditional data carrying capacity. It is obtained by multiplying the utilization by the transmission time of a packet, which is, in our case, equal to *1*.

Although the model is easy to use, and gives satisfactory results when the network is lightly loaded, Armyros [5] has shown that it is not appropriate when the network is heavily loaded. Thus, we are proposing a new model. The essence in our approach is to treat *m(n)* as the average utilization according to Sohraby's model [4].

In order to do this, the first step is to define the mapping between the two incompatible presentations. In Almes and Lazowska's model, the instantaneous load is the function of the number of packets in the system, while we have to define it as the throughput in Sohraby's model. The simple mapping we are using is based on the following idea. The instantaneous load due to the retries is proportional to the number of stations in the system, *n*, the contention time, and inversely proportional to the time between two attempts by the same station. The maximum cycle length occurs when the station makes 16 attempts and then drops the packet. It takes 7176 cycles, which divided by 16 attempts gives us the average time between retries, *RT*, of 225 slot times. However, to obtain this cycle time, the collision probability must be equal to one. In a more realistic high load case when collision probability is 3/4, the expected delay divided by the expected number of retransmissions is roughly equal to 100 slots. We are taking this number as the first approximation in the following way.

$$IL = n \times \frac{ST}{RT} \approx n \times \frac{a}{100 \times a} = \frac{n}{100} \quad \text{(EQ 10)}$$

In other words, the model will have 100 states, with a population of one station corresponding to 1 percent of the total load. Having defined this, we can enter the Sohraby's model, to find the expression for the utilization of the channel.

The rate at which the individual packets depart the system will be equal to the ratio between the number of successfully departed and the total number of packets, *S/G* obtained from this model. This number is further equal to *1/r*, and leads us to an intuitive justification of this approach saying that from every r attempts to transmit, one will succeed. The departure rate will be:

$$through(n) = \frac{1}{r\left(\frac{n}{100}\right)} \quad \text{(EQ 11)}$$

So, in our model the function *m(n)* will be equal to *through(n)*, while the function *l(n)* will be constant function *load* depending only on the input rate, as in the original model of Almes and Lazowska.

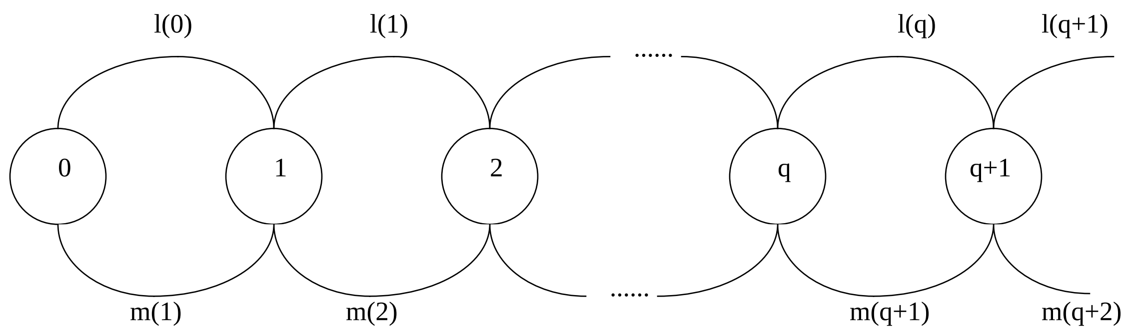
Using the standard approach in solving the death-birth process, the state probability distribution is given by the following formula:

hosts, waiting for transmission over the Ethernet, will force us to omit the other functional dependencies.

2.3.1 The Model

In general, a load dependent service center takes a form of one-dimensional continuous time Markov chain, with the transition rates dependent on the load on the network. The states represent the population of the network, and only transitions between the neighbouring states can take place (birth-death system). Transition rates between the states depend on the instantaneous load on the networks.

FIGURE 5 Markov chain model



In the specific case of Almes' and Lazowska's application of this model, they used the approximate capacity formula for Ethernet of Metcalfe and Boggs [2] to determine the death rate. The assumption there was that with Q active users, the network would alternate between "contention periods" (similar to optimally controlled finite population slotted Aloha) and "packet transmissions". Under these assumptions, Metcalfe and Boggs found that the utilization is given by

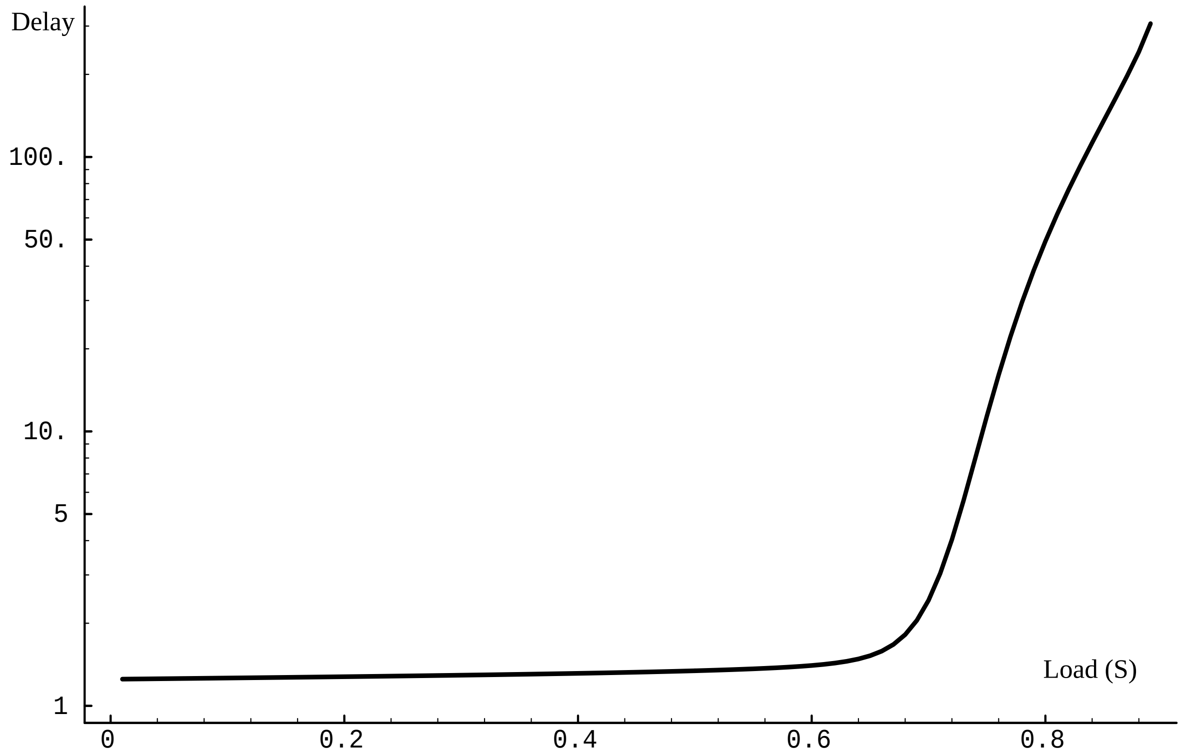
$$S(Q) = \frac{1}{1 + W(Q) \bullet 2a} \quad \text{(EQ 7)}$$

where time is normalized, so the length of the packetis unity, *2a* is the contention slot time and *W(Q)* is the mean number of contention slots in the contention period. They produced the expression for obtaining this value. From this result, the transition rates are derived. They are described by the following two equations:

$$l(n) = l \quad \text{(EQ 8)}$$

Since this function grows very fast after some load, next figure shows the delay in a logarithmic scale.

FIGURE 4 Load-delay characteristics- logarithmic scale



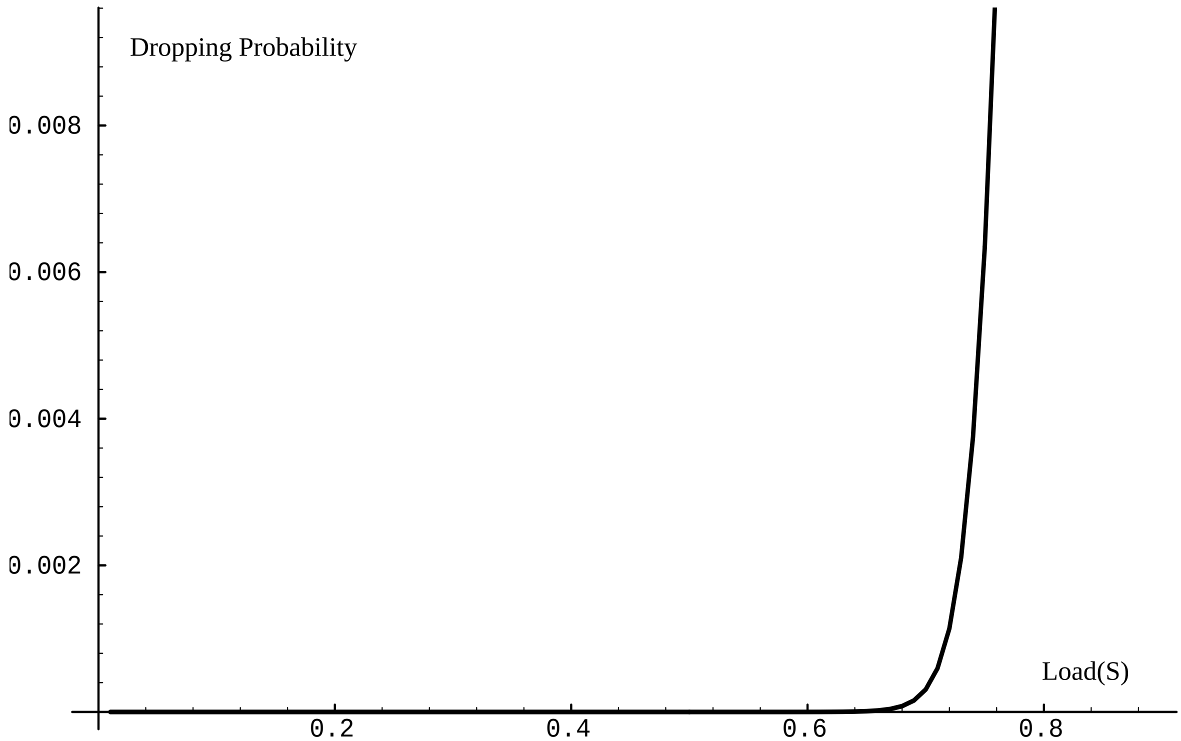
Its shape matches the shape of the curve obtained in Armyros' research [5].

2.3 Queueing Model

The analytical model given in previous section is useful when modelling a single Ethernet network in isolation with uncorrelated arrivals generated by a large number of users. In reality, however, the Ethernet will in general be used to connect a limited number of users exchanging packets according to multi-packet transactions.

Almes and Lazowska [1] introduced a simple model of the Ethernet, suited to incorporation as a service center in a queueing network model of a distributed system. In queueing network modelling terminology, this type of model is known as a *load dependent service center.* The main idea behind this concept is to describe the situations when the service time in some service center in a queueing network depends on the queue size. This situation happening in Ethernet networks and our task here is to determine how the service rate depends on the queue size from the knowledge of protocol operation. However, the fact that method only allows a dependence on the total number of packets at all to 0.1 when the collision probability is 0.87. From the previous curve, it will happen when the offered load is 0.88. Figure 3 shows the dropping rate as a function of load on the network. It is obvious that the knee in curve happens at the point after the Ethernet will drop the substantial share of the packets.

FIGURE 3 Probability of Dropping a Packet as a Function of Load



By knowing the number of retries, we can further calculate the expected delay in the system.

2.2 The Expected Delay

We can now easily calculate the expected retransmission delay using the following formula

$$E(backdelay) = \sum_{i=1}^{16} P\{Ncoll = i\} \times Delay(i) \quad \text{(EQ 5)}$$

where the probability that there where exactly *i* attempts is given by the geometric distribution, and the function *Delay* depends on the type of the backoff algorithm used. In our example, it is given by:

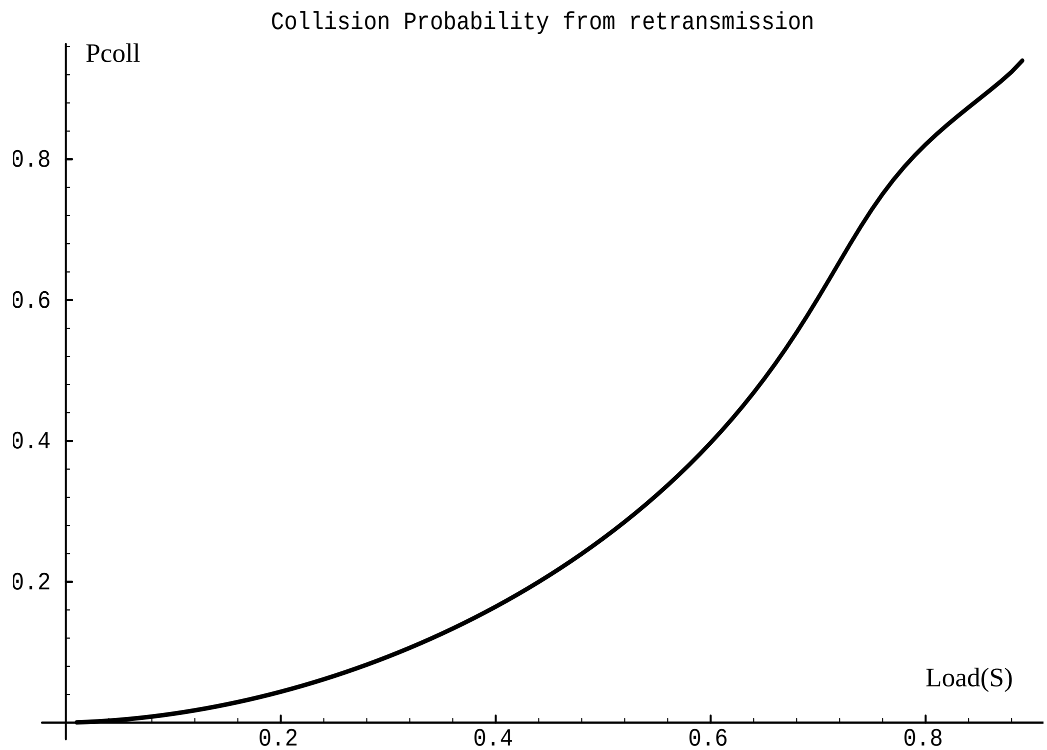
$$Delay(i) = 2^{(i-1)} \quad \text{(EQ 6)}$$

Given an assumption that the number of retransmission has a geometrical distribution, we can extract its parameter by matching means. It is described by the following formula:

$$p_{coll}(r) = 1 - \frac{1}{r} \quad \text{(EQ 4)}$$

Figure 2 shows the graph of the probability of success in the same examples in previous figure.

FIGURE 2 Collision probability derived from number of retransmissions



Implicit in the geometric distribution is the assumption that unsuccessful packets can potentially be retransmitted an unlimited number of times. However, according to the Ethernet protocol specification, the packet will be dropped from the system after 16 collisions. Therefore, before accepting the geometric assumption, we should calculate the probability that more than 16 attempts is required, namely $p_{coll}(r)^{16}$. As long as this "overflow" probability is small, our geometric assumption should give acceptable results.

In the area of low and medium loads, this probability will be held small. With the collision probability rising with the load, the dropping rate will increase. In the previous example, when the offered load is 0.75, then the collision probability is approximately 0.75. According to our assumption of the geometrical distribution, the probability that the packet will be dropped after 16 attempts is 0.01. The dropping probability will rise method is often called the *G-S* method. In solving this model, Sohraby uses the embedded Markov chain process description of the system. The signal propagation time is defined to be *a*, the collision duration is *b*, and the packet transmission is equal to *1*.

The Model produces the solution as a function of the form

$$S = f(G) \quad \textbf{(EQ 1)}$$

On the other hand, the total traffic is linked with the successfully transmitted packets via factor *r*, i.e.

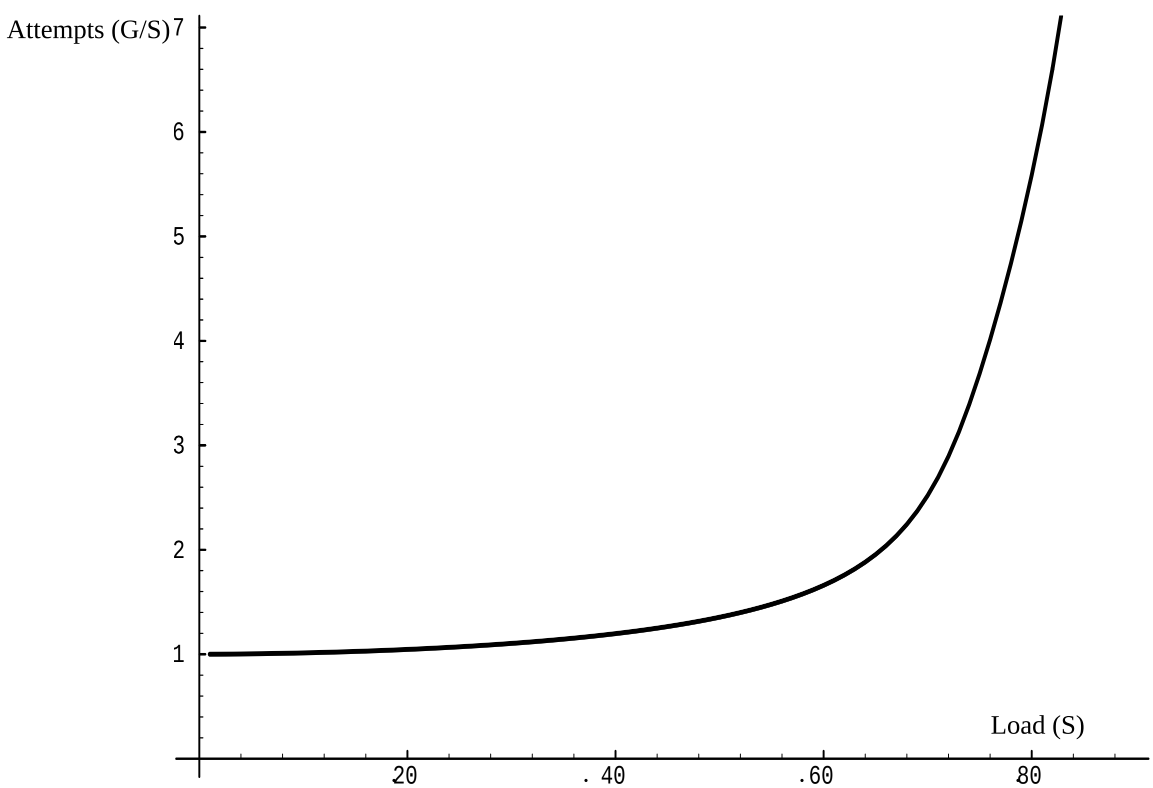
$$G = r \times S \quad \textbf{(EQ 2)}$$

In order to calculate the mean number of retransmissions, we will solve this system by finding the fixed point of the system

$$S = f(r \times S) \quad \textbf{(EQ 3)}$$

for all possible effective loads *S*. Here, we assume that system is stable, which is not true above some maximum sustainable load. The following figure shows one graph of the retry attempts, with the parameters a=0.022 and b =0.2 a

FIGURE 1 The number of transmission attempts as a function of network load in percents



CHAPTER 2

Approximate Methods for Incorporating the Backoff Algorithm in CSMA/CD

This chapter describes two approximate models that include the effects of the binary exponential backoff algorithms on CSMA/CD multiple access scheme. The first one is the extension of the infinite population model of Sohraby et al. The second is similar to the approach used by Almes and Lazowska and gives us a representation of Ethernet as a load-dependent service center in a product-form queueing network model.

2.1 The Infinite Population Model

Our goal in this section is to produce an approximate delay-throughput response time curve for an infinite population CSMA/CD system that includes the effects of the backoff algorithm. Our approach is to obtain the collision probabilities as a function of throughput from Sohraby's model [4]. This model provides us with the relationship between throughput, *S*, and the offered load, *G*, from which we extract the expected number of retransmissions per packet, *G/S-1*. Then, we assume that the number of retransmissions has a geometric distribution, for which we can extract the probability of the failure in any attempt, p_{coll}. After that, we can determine the mean backoff delay as a function of the retransmission number distribution and the retry policy. This will give us a simple delay model of CSMA/CD networks which includes the backoff algorithm. Details and the simplifying assumptions used will be given in following sections.

2.1.1 Extension to Sohraby's Model

In their paper [4], Sohraby et al. gave a detailed model of the infinite population CSMA/CD network. The basic assumption is that the packets carrying the original traffic and the retransmission packets are together forming a Poisson stream of packets. The process of generation of all the packet transmission attempts is characterized by the Poisson parameter *G*, while the throughput is given by the factor *S*. That is the reason why this

In solving the exact backoff model, very fast machines are needed. Our suggestion is that probably the massively parallel computers, will be capable to deal with the realistic two station backoff model.

In order to calculate the delay caused by the backoff, the mean number of retries is needed. This number can be obtained from the characteristics which show the utilization of the channel as a function of the total traffic (including retries). This functionality is described in Sohraby's model. From this information, the expected number of retransmissions is obtained. By assuming that the probability of a success is the same at every attempt, we are led to a geometric distribution for the number of attempts. Combining this with the known distribution for the backoff delay at each attempt, we can calculate the delay encountered in the process of conflict resolution, i.e., competition for the channel.

1.1.2 Queueing System Modelling

The usual method of characterizing Ethernet in a queueing networks model [1] was shown to be inappropriate under the heavy load in [5]. Almes and Lazowska's approach was to define the birth-death process model of the number of the packets in the system. All the transitions are the function of the number of the packets in the system, i.e. its state. They were using a rough estimate of the throughput [2] to model the transitions, while our idea is to use the better function for the same purpose. We are using the throughput obtained from Sohraby's model and our load-delay characteristics show the better results, especially when the loads are higher.

1.2 Modelling Two Stations

Since all the previously considered models are still only rough approximations, our intention here was to produce a more accurate model for the two station case.

Our motivation here was two-fold. First, we hope to produce a model that captures the qualitative behaviour of the general case. Second, we believe the two station model itself is of practical importance due to the introduction of high performance multiport bridges ("switches").

The last chapter presents one such model. The general characteristics of any such model will be the abundance of the states, and we made a lot of efforts to keep the state space as low as possible. Even with the biggest possible approximations, the realistic model will havemore than 10000 states, which makes it impractical. The results presented hereare limited to small buffers (with capacity of two packets), and the backoff algorithm modelled uses the backoff counter in modulo two number system.

1.3 Conclusion

The work presented here proves that it is possible to model the backoff algorithm in CSMA/CD networks, but that the cost for doing this can be enormous. A few ideas in implementing particular approaches to different modelling methods, not mentioned here. Results are described in the concluding sections of every Chapter.

CHAPTER 1 Introduction

The following chapters describe roughly three different parts of the work. The *approximate backoff modelling* includes the extension of the classical results in unbuffered Ethernet as well as the extension of the queueing service center model. The main result of this report is the *detailed model* of the backoff algorithm in CSMA/CD networks, for a case of two-station system.

1.1 Backoff Algorithm Approximate Modelling

Chapter 2 contains the description of the results in the area of the approximate modelling of the backoff algorithm. Our basic idea was to provide simple models which will take into account the effects caused by the exponential backoff algorithm. Two approaches are presented, each of them useful in different modelling technique domain.

Since Sohraby's model [4] describes the infinite population Ethernet with non-buffered stations, our first attempt was to incorporate the backoff effects into this model. The final model is not much more complex than the original one, but shows the impact of the backoff algorithm to performance of highly loaded networks and offers the load-delay characteristics, not too often found in the Ethernet analysis results.

The second model is aimed to applications in domain of *queueing network modelling* [3]. The existing models have been shown to be inappropriate for high load, so our work was aimed to investigate a way to improve the model in this mode of operation.

1.1.1 Approximate Infinite Population Models

The existing models of Ethernet do not deal with the backoff algorithm. The reason is in the complexity of the model encountered when trying to incorporate the effect of the backoff. We tried to make the simple extension in order to incorporate the missing part.

Abstract

The medium access control protocol in Ethernet systems consists of two parts: the 1-persistent CSMA/CD algorithm describes rules that a ready station must follow while trying to acquire the medium, while the binary exponential backoff algorithm determines how long a ready station must wait before retrying after each failed attempt. Although both algorithms are equally important in determining the actions of each station, previous work on modelling the performance of Ethernet has focused on the effects of 1-persistent CSMA/CD. In this work, we describe three performance models that do incorporate the effects of the backoff algorithm. First, we consider the domain of product-form queueing networks, where we modify the well-known representation of Ethernet as a load-dependent service center by Almes and Lazowska to incorporate a more realistic service rate function. Second, we consider the classical delay-throughput model of Sohraby et al. We show that it can be augmented with a simple auxiliary model of the retransmission delays, in which the number of attempts is geometric and the mean interattempt times are determined by the backoff algorithm, to give a good estimate of the response times except at very high loads. And finally, we consider a detailed embedded Markov chain model of the two station case. Although some useful results were obtained with a truncated model (where we imposed unrealistically small limits on the queue size at each station and on the maximum number of retransmissions per packet), we believe that the complexity of this approach renders it impractical. For the same computational effort, better results can be obtained via simulation.

Modelling the Exponential Backoff Algorithm in CSMA/ CD Networks

Zeljko Zilic and Mart L. Molle

Technical Report CSRI-279
University of Toronto
Computer Systems Research Institute

Modelling the Exponential Backoff Algorithm in CSMA–CD Networks

Željko Žilić and Mart L. Molle

Technical Report CSRI-279
October 1992

Computer Systems Research Institute
University of Toronto
Toronto, Canada
M5S 1A1

The Computer Systems Research Institute (CSRI) is an interdisciplinary group formed to conduct research and development relevant to computer systems and their application. It is an Institute within the Faculty of Applied Science and Engineering, and the Faculty of Arts and Science, at the University of Toronto, and is supported in part by the Natural Sciences and Engineering Research Council of Canada.