A Numerical Analysis of Gallager's Contention Resolution Algorithm

Gallager proposed a contention resolution protocal of multiaccess network in 1978 [1]. The algorithm can be written as follow:

Algorithm 1 Gallager's Algorithm

```
procedure ConflictResolutionPhase(interval i)
    divide i into two halves i_1 and i_2
    try sending all packets in i_1
    if hear silence then
        mark i_1 as dealt
        ConflictResolutionPhase(i_2)
    else if hear success then
        mark i_1 as dealt
        try sending all packets in i_2
        if hear success then
            break
        else                                    ▷ it's not possible to hear silence here
            ConflictResolutionPhase(i_2)
    else
        ConflictResolutionPhase(i_1)
procedure Gallager
    repeat
        pick new enabled interval i from undealt time axis
        try sending all packets in i
        if hear silence or success then
            mark i as dealt
        else if hear conflict then
            ConflictResolutionPhase(i)
    until all packets are sent
```

With the assumption that new packets arrive as a Poisson point process, which implies that the number of packets inside each interval obeys Poisson distribution, we analyze the performance of the algorothm by measuring the throughput of the network, which is defined to be the expected number of packets delivered per time unit. The throughput can be computed as the expectation of the number of packets delivered divided by the time spent to deliver these packets.

Suppose the number of packets inside each initial enabled interval is a Poission random variable with expectation λ. Our goal is to find the optimal λ such that the throughput of the protocol is maximized. To compute the throughput as a function of λ, we define the following quantities:

- $p1(\lambda)$: suppose X, Y are independent Poisson random variables with expectation $\lambda/2$,

$$p1(\lambda) \stackrel{\text{def}}{=} \Pr\left[X = 0 \mid X + Y \geq 2\right] = e^{-\lambda/2}(1 - (1 + \lambda/2)e^{-\lambda/2}) / \left(1 - (1 + \lambda)e^{-\lambda}\right)$$

- $p2(\lambda)$: suppose X, Y are independent Poisson random variables with expectation $\lambda/2$,

$$p2(\lambda) \stackrel{\text{def}}{=} \Pr\left[X = 1 \mid X + Y \geq 2\right] = (\lambda/2)e^{-\lambda/2}(1 - e^{-\lambda/2}) / \left(1 - (1 + \lambda)e^{-\lambda}\right)$$

- $p3(\lambda)$: suppose X, Y are independent Poisson random variables with expectation $\lambda/2$,

$$p3(\lambda) \stackrel{\text{def}}{=} \Pr\left[X \geq 2 \mid X + Y \geq 2\right] = 1 - p1(\lambda) - p2(\lambda)$$

- $p4(\lambda)$: suppose X is a Poisson random variables with expectation $\lambda/2$,

$$p4(\lambda) \stackrel{\text{def}}{=} \Pr\left[X = 1 \geq X \geq 1\right] = (\lambda/2)e^{\lambda/2}/(1 - e^{-\lambda/2})$$

- $T(\lambda)$: the expected time to resolve an enabled interval (the expected time between from entering an enabled interval to finishing the conflict resolution phase), where the number of packets in the interval is a random variable $X \sim \text{Poisson}(\lambda)$.
- $t(\lambda)$: the expected time to resolve an enabled interval, where the number of packets in the interval is a random variable $X \sim \text{Poisson}(\lambda)$ and is known to be at least 2.
- $N(\lambda)$: the expected number of packets successfully delivered in one enabled interval, where the number of packets in the interval is a random variable $X \sim \text{Poisson}(\lambda)$.
- $n(\lambda)$: the expected number of packets successfully delivered in one enabled interval, where the number of packets in the enabled interval is a random variable $X \sim \text{Poisson}(\lambda)$, and is known to be at least 2.

These quantities satisfy the following relations:

$$T(\lambda) = 1 + (1 - (1+\lambda)e^{-\lambda}) \cdot t(\lambda) \tag{1}$$

$$t(\lambda) = 1 + p1(\lambda) \cdot t\left(\lambda/2\right) + p3(\lambda) \cdot t\left(\lambda/2\right) + p2(\lambda) \cdot \left(1 + (1 - p4(\lambda)) \cdot t\left(\lambda/2\right)\right) \tag{2}$$

$$N(\lambda) = \lambda e^{-\lambda} + (1 - (1+\lambda)e^{-\lambda}) \cdot n(\lambda) \tag{3}$$

$$n(\lambda) = p1(\lambda) \cdot n\left(\lambda/2\right) + p3(\lambda) \cdot n\left(\lambda/2\right) + p2(\lambda) \cdot \left(1 + p4(\lambda) + (1 - p4(\lambda)) \cdot n(\lambda/2)\right) \tag{4}$$

We have the recursion in equation (2) because: when it is known that at least 2 packets is in the interval,

- With $p1(\lambda)$ probability, we have 0 packets in the first half of the interval. We recurse the algorithm on the second half of the interval.
- With $p3(\lambda)$ probability, we have 2 packets in the first half of the interval. We release the second half of the interval and recurse the algorithm on the first half.
- With $p2(\lambda)$ probability, we have 1 packet in the first half of the interval. We send the packet in the first half of the interval. With probability $p4(\lambda)$, the second half of the interval only has one packet. So with $(1 - p4(\lambda))$ probability, we recurse the algorithm on the second half of the interval.

Similarly we obtain the recursive definition of (4).

By linearity of expectation, when the protocol runs for long enough, the throughput is equal to $\frac{N(\lambda)}{T(\lambda)}$. We computed the numerical value of the throughput with the following Mathematica code:

```
p1[lam_] := Exp[-lam/2]*(1-(1+lam/2)Exp[-lam/2])/(1-(1+lam)Exp[-lam]);
p2[lam_] := (lam/2)*Exp[-lam/2]*(1-Exp[-lam/2])/(1-(1+lam)Exp[-lam]);
p3[lam_] := 1-p1[lam]-p2[lam];
p4[lam_] := (lam/2*Exp[-lam/2])/(1-Exp[-lam/2]);
t[lam_] := If[lam<0.00001, 0, 1+p2[lam]+(p1[lam]+p3[lam]+p2[lam](1-p4[lam]))*t[lam/2]];
tt[lam_] := 1+(1-(1+lam)Exp[-lam])t[lam];
n[lam_] := If[lam<0.00001, 0, (p1[lam]+p3[lam]+p2[lam](1-p4[lam]))*n[lam/2]+p2[lam]*(1+p4[lam])];
nn[lam_] := lam*Exp[-lam]+(1-(1+lam)Exp[-lam])*n[lam];
f[lam_] := nn[lam]/tt[lam];
NMaximize[f[lam], lam]
   {0.487116, {lam -> 1.26636}}
```

Note that in the code above, as the base case of the recursion, t and n are considered 0 when λ is below some threshold (0.00001 is used in the code above).

The output of code above agrees with the result provided in Gallager's original paper.

We further plot the expected throughput of different λ around the optimal value, which is shown in Figure 1.

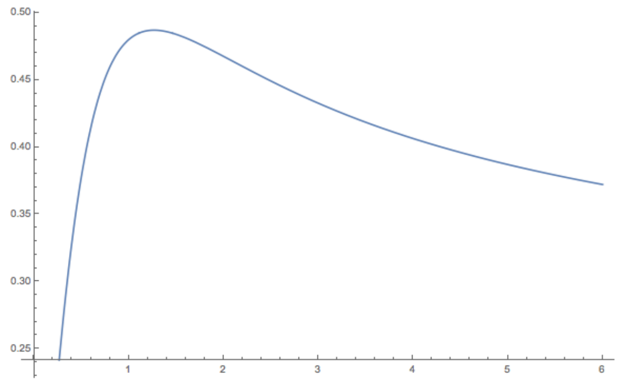

FIGURE 1. Performance of Gallager's Algorithm under Different λ

References

[1] R. G. Gallager, *Conflict resolution in random access broadcast networks*, Proc. AFOSR Workshop Commun. Theory Appl. (Provincetown, MA), 1978, pp. 74-76.