

## Regarding $\lambda_{BO}$ and $q_0$

$\lambda$  is the arrival rate of packets at individual nodes in the real time (*System Time*) as we have referred to it. Our analysis has been done by ignoring the channel activity periods, i.e., in *Backoff Time* only. Hence any packet arrivals happening during the channel activity period will be ignored. Refer to the Fig. 3a periods 3, 5, and 7. (each period is separated by a vertical dotted line, and first period is between the solid vertical line and first dotted line). The arrivals marked with a “**x**” happen during channel activity periods and have to be accounted for. In order to account for these arrivals,  $\lambda$  has to be scaled up.

Hence, the relation between  $\lambda$  and  $\lambda_{BO}$  can be summarized as follows:

Given  $M$  packet arrivals in  $(x + y)$  slots of System Time, where  $x$  slots are in Backoff Time, and  $y$  slots are in channel activity. Then, the following holds,

$$\begin{aligned}\lambda &= M/(x + y) \\ \lambda_{BO} &= M/x \\ \therefore \lambda_{BO} &= \lambda \cdot \frac{(x + y)}{x}\end{aligned}\tag{a}$$

For this particular equation in the paper, there is a typo error. The results remain the same. Eq (4) in the paper should have been

$$scalingFactor = \frac{1}{\left(1 - (1 - \beta)^{n^*}\right) \left(T_c \gamma + T_s (1 - \gamma)\right) + 1}\tag{b}$$

$$\therefore \lambda_{BO} = \frac{\lambda}{scalingFactor}\tag{c}$$

From Eq (a), (b) and the set of equations preceding (4) in Section 3.2 of the paper, (c) can be derived.

Here, it can be noted that for scalingFactor, denominator is always greater than 1. Hence, scalingFactor is less than 1 and  $\lambda_{BO}$  is greater than  $\lambda$ , i.e., scaled up to account for arrivals during channel activity periods.

As already noted in Remarks before Section 3.3 of the paper, in order to get the equations to converge, we first fix  $q_0$  and iterate between (1) and (2). Once the values for  $\beta(q_0)$  and  $\gamma(q_0)$  converge,  $q_0$  is updated using (5) and the procedure is repeated till all values converge till the 3<sup>rd</sup> decimal point.