Performance Analysis of IEEE 802.11

Department of Electrical and Computer Engineering
Rutgers University, Piscataway, NJ 08854

Abstract

1. Introduction

Existing work includes [Bianchi 2000], and some problems that we identify with this work are:

- It considers the ***probability*** of packet transmission within a backoff slot; it would be more of interest to start with the number of nodes queued for transmission and derive collision and success probabilities
- It specifies the collision probability for *any* time slot rather than considering the actual period of vulnerability
- It does not use Poisson arrival model for packet arrivals

The approach presented below considers the ***rate*** of transmissions per packet per node per second[1]. Its main characteristic is that it gives an idea of system performance not only in terms of collision probability, but also in terms of number of mobile stations and their traffic load. As in [Bianchi 2000], we also require "saturated" traffic condition.

Our objective is to derive the probabilities of transmission success or collision, and from these we can derive link throughput and delay bounds. The system can be represented as in Fig. 1.

m = total number of nodes

n = nodes in **queuing state** 1 or higher ($n \leq m$), i.e., these are backlogged nodes
$= fct(m, \lambda_{Appl})$

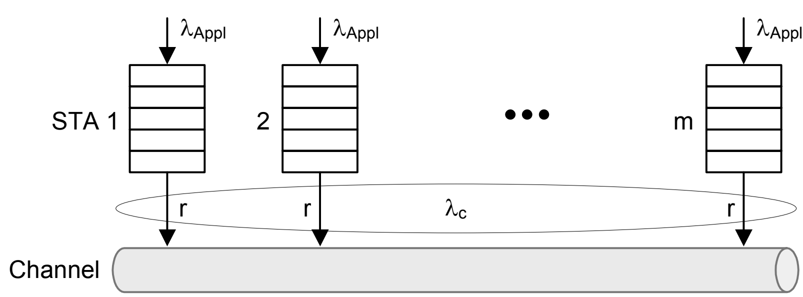

Fig. 1. System representation of *m* stations competing for a single wireless channel. Aggregate channel arrival rate λ_c.

[1] *Rate* carries much less information than *probability*; thus it is easier to obtain. Given a probability, we can determine the rate; also, probability is usually associated with probability distribution function (pdf), which specifies a random process.

where λ_{Appl} is the rate at which the node application generates new packets. For simplicity, we assume that all nodes run the same application, so all nodes have the same λ_{Appl}. Our first goal is to derive the channel arrival rate λ_c. We work on per node basis to get to the total aggregate rate on the wireless link. In summary,

num. of backlogged nodes → probability of backoff stage → offered rate per node → aggregate channel rate

The analysis will be performed for the steady state—we assume that the average number of backlogged nodes remains constant. Although at one time a node may enter an empty state from backlogged state, there will be another node transitioning in the opposite way, so that the system is maintained in a balanced state, with constant $E[n] > 0$. [I'm not sure if non-zero n is important]

As a consequence of non-zero n, the channel is working in a saturated traffic condition. That is, since at anytime there is at least one backlogged node, the channel is always "busy"—from the channel point of view packets always arrives regardless from which node. (Note that saturated channel does not imply saturated stations.)

Finally, we assume that all nodes with a packet to transmit have the same priority to use the channel.

2. Channel Arrival Rate

In CSMA, from a station's viewpoint, the channel can be idle or busy. When the channel is busy, the station's transmission algorithm is at a halt; conversely, when the channel is idle, the ready stations contend for the channel. In 802.11, the *contention period* comprises an integer number of countdown slots. The countdown is set randomly and when it reaches zero, the station transmits. Three types of channel events are possible during the contention period: idle, successful transmission, and collision. The probabilities associated with these events are denoted as p_{idle_cp}, p_{succ_cp}, and p_{coll_cp}, respectively. If the channel becomes busy during the countdown, the station freezes and continues the countdown only when the channel again becomes idle.

When the station's countdown reaches zero, it transmits and the channel becomes busy. However, other stations do not know this for a short period due to the finite speed of signal propagation and detection. This is called *vulnerable period* and during it there are only two possible events: success and collision, with the probabilities p_{succ_vp} and p_{coll_vp}, respectively.

We derive the channel arrival rate by looking into the periodicity of events from a station's viewpoint. The rate (frequency) of events is then the reciprocal of the periodicity.

2.1. Event Probabilities

Assume a unit time during which a station can transmit at most one packet. We can consider the system as consisting of n independent Bernoulli trials. The probability that i out of n backlogged nodes transmit a frame with probability q during the unit period is:

$$Q(i, n) = \binom{n}{i} \cdot q^i \cdot (1-q)^{n-i}$$

Then, the probability of an idle slot during the contention period is:

$$p_{iddle\_cp} = \binom{n-1}{0} \cdot (1-q_c)^{n-1} \tag{1a}$$

The probability of a successful transmission during the preemption in a contention slot is:

$$p_{succ\_cp} = \binom{n-1}{1} \cdot q_c \cdot (1-q_c)^{n-2} \tag{1b}$$

And, the probability of a collision during the preemption in a contention slot is:

$$p_{coll\_cp} = 1 - p_{iddle\_cp} - p_{succ\_cp} \tag{1c}$$

The probability of a successful transmission during the vulnerable period is the probability that exactly one node transmits during the vulnerable period (the one whose vulnerable period is being considered):

$$p_{succ\_vp} = \frac{\Pr(\text{only one node transmits})}{\Pr(\text{at least one node transmits})} = \frac{\binom{n}{1} \cdot q_v \cdot (1-q_v)^{n-1}}{\binom{n}{0} \cdot (1-q_v)^{n}} \tag{2a}$$

Conversely, the probability of a collision in the vulnerable period is:

$$p_{coll\_vp} = 1 - p_{succ\_vp} \tag{2b}$$

The probabilities q_c and q_v that a station transmits during the respective periods are derived below.

The above analysis considers the channel arrival process as Poisson. The intuition is that the backoff mechanism randomizes waiting time between retransmissions, so that Poisson assumption for the frame interarrival time is valid (i.e., for the times between the packet transmissions). If the system appears memoryless, we can use Poisson interarrival assumption. For average, long-term first order statistics, Poisson assumption should be OK; we can consider the bursty traffic case later. [For self-similar processes, there is no natural length for a "burst," traffic burst appear on a wide range of time scales.] Our system is (approximately) memoryless in terms of offered traffic load—the fact that the previous time slot was idle does not change the probability that the next slot will be idle.

2.2. Individual Transmission Rate

We start by deriving the channel transmission rate from the viewpoint of a single node. We assume that transmission is error-free and consider only packet collisions. The arrival events for a packet are the actual arrival to the station's MAC and a repeated "arrival" due to a transmission failure. The departure events are the packet transmissions, which can end in success or collision. The arrival rate r per node is the reciprocal of the expected duration of the arrival-departure period. This is illustrated in Fig. 2. Channel idle period includes the DIFS period and exponential backoff countdown period. Since every packet transmission is preceded by DIFS, the inter-event contention period (CP) does not include DIFS[2].

[2] Strictly speaking this is not true, since for a non-backlogged station a new packet that arrives during the idle channel can experience transmissions by other stations even during its DIFS period. Also, if we were to consider *erroneous* channel, a station that receives a corrupted frame precedes the next frame by EIFS and this should be factored in into its CP.

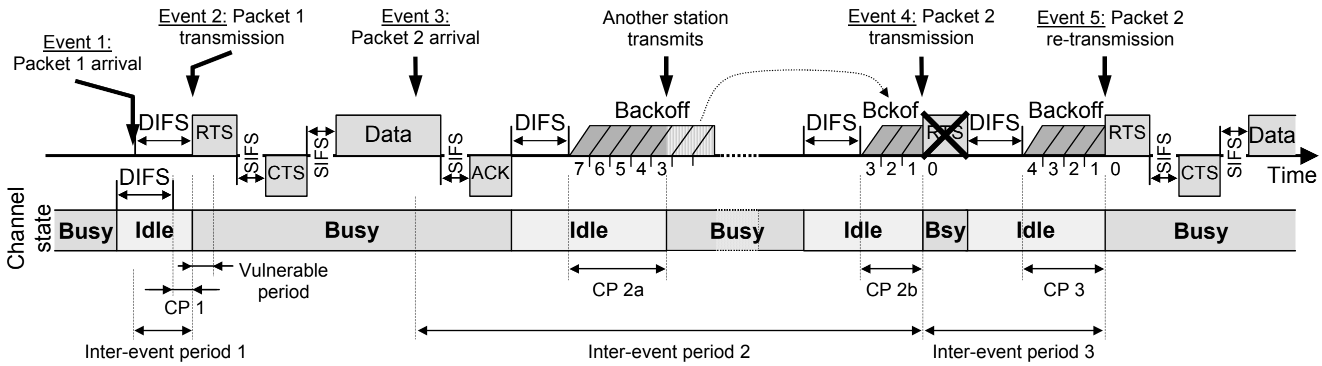

Fig. 2. Definition of the inter-event period and the contention period (CP). Note that the CP 2 consists of two components. Also, if a packet arrives during an idle channel (as is the case for packet 1), it waits only for DIFS time, i.e., there is no backoff period. In this case, the CP overlaps with its DIFS.

The arrival rate r per node is:

$$r = E[\text{rate of frame transmissions}] \text{ frames/sec} = \frac{1}{E[\text{periodicity of events}]} = \frac{1}{t_{ep}} \qquad (3)$$

where t_{ep} is the expected duration of the period between two events at a node.

If other node(s) transmit before this node counts down to zero, the CP will consist of multiple segments, as shown for CP 2 in Fig. 2. The preempted node will continue counting down from where it left off, rather than selecting a fresh exponential backoff count. Also, CP is relevant only for the n backlogged nodes. Once a node has no packets queued for transmission, it belongs to the $(m - n)$ unbacklogged nodes and its CP is undefined.

IEEE 802.11 defines the retry counter limit, so the packet is discarded after a certain number ℓ of retransmission attempts. It is common to assume ℓ=7, as stipulated by the standard. The **backoff stage** is the round of re-transmission $[0..\ell]$, which determines the contention window size. The contention window $CW(k)$ grows between CW_{min} = 32 and CW_{max} = 1024. Thereafter, for the backoff stages $k > \log_2(CW_{max}/CW_{min}) = 5 < \ell$ the backoff window remains constant. Since backoff counts are drawn according to a uniform distribution, the expected number s_k of slots in backoff stage k is:

$$s_k = \sum_{j=0}^{CW(k)-1} \frac{1}{CW(k)} \cdot j \approx \begin{cases} 2^{k-1} \cdot CW_{\min}, & \text{for } 0 \le k \le 5 \\ \frac{CW_{\max}}{2}, & \text{for } 5 < k < \ell \end{cases}$$

The events that affect the node's backoff stage k occur only during the vulnerable period[3] of $t_{vp} = t_{RTS} + t_{SIFS} = 170\ \mu$s for IEEE 802.11b DSSS physical layer protocol, see illustration in Fig. 3(a).

[3] Strictly speaking, this is correct only if we assume that all stations can hear each other. Otherwise, for hidden stations, the vulnerable period is until the next CTS frame. Several conflicting definitions of this period must be reconciled. At some places we say it includes SIFS after RTS because of hidden stations. At other places, hidden stations are not included. Finally, in Figure 2, vulnerable period is shown as if it is only one slot long.

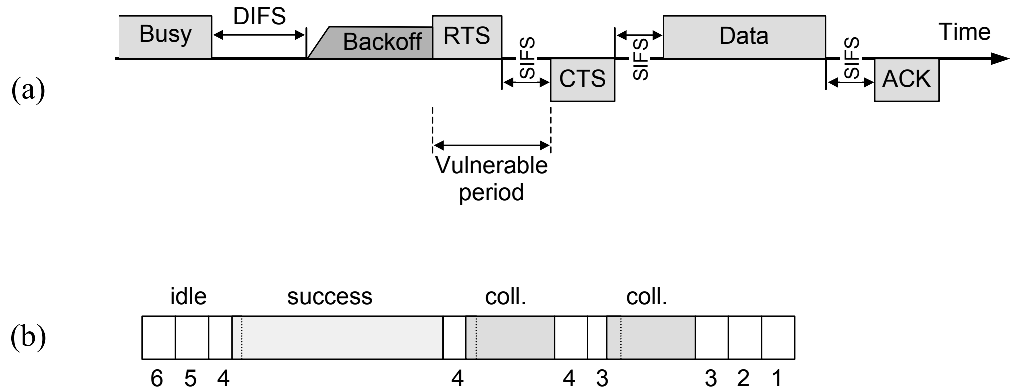

Fig. 3. (a) Definition of the vulnerable period. Note that SIFS period after RTS belongs to the vulnerable period because of the hidden station problem. (b) A station may be preempted multiple times, even within the same slot, e.g., slot 4, during the countdown.

[Why the vulnerable period only includes one RTS and not two? If another station transmits RTS right before the considered station, collision will also happen.]

The time t_{ep} is a function of the number of backlogged nodes n and the node arrival rate, $t_{ep} = fct(n, r)$. Let $t_{start}(k)$ denote the period from the start of the k^{th} backoff stage until the attempted transmission at that stage. Then, $t_{start}(k)$ = DIFS + $t_{bckoff}(k)$, where the time spent in backoff countdown is:

$$t_{bckoff}(k) = E[\text{Number of contention slots at stage } k] \times E[\text{cont. slot length}] = \left(s_k \cdot \frac{1}{p_{idle\_cp}} \right) \times t'_{slot}$$

where t'_{slot} is the expected contention slot length. The scaling factor $1 / p_{iddle\_cp}$ amends the average number of backoff slots s_k for the repeated counting of the busy contention slots, see Fig. 3(b). Contention slots are either idle or busy, so if p_{iddle_cp} is the fraction of idle slots, then $1 / p_{iddle\_cp}$ is the total number of contention slots.

If a transmission attempt suffers a collision during the vulnerable period t_{vp}, the total time[4] spent for this try is $t_{coll} = t_{vp}$ = RTS. Conversely, if a transmission attempt succeeds, the total time is t_{succ} = RTS + SIFS + CTS + SIFS + $E[P]$ + SIFS + ACK, where $E[P] = \frac{\text{Average frame length}}{\text{Transmission rate}} \left[\frac{\text{bits}}{\text{b/s}} \right]$ is the expected duration of the data frame transmission. Then,

$$t'_{slot} = t_{slot} \cdot p_{idle\_cp} + t_{succ} \cdot p_{succ\_cp} + t_{coll} \cdot (1 - p_{idle\_cp} - p_{succ\_cp})$$

which is independent of the node's backoff stage. We also need to account for the initial waiting time, upon the packet's arrival, before the first transmission. This is the residual time until the end of the channel busy state. The channel may be busy due to a successful transmission or a collision:

$$E[R] = ½ \, (p_{succ\_vp} \cdot t_{succ} + p_{coll\_vp} \cdot t_{coll})$$

Let b_k denote the probability of node being in backoff stage k. Then,

[4] Note that propagation delays are ignored, but can be included for accuracy.

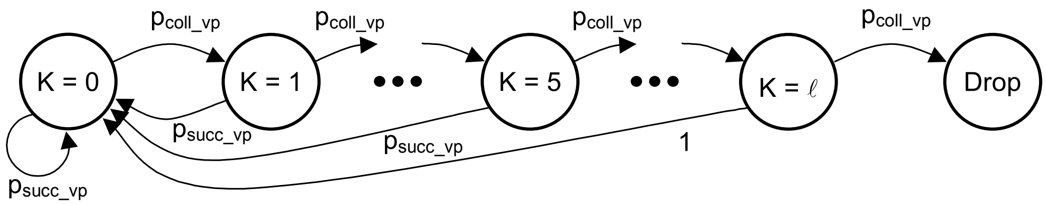

Fig. 4. Markov chain representation of the random process *K* counting a node's exponential backoff stage.

$$
\begin{aligned}
t_{ep} &= b_0 \cdot \left[E[R] + t_{start}(0)\right] + \sum_{k=1}^{\ell}\left(b_k \cdot t_{start}(k)\right) \\
&= \text{DIFS} + b_0 \cdot \left(\frac{p_{succ\_vp} \cdot t_{succ} + p_{coll\_vp} \cdot t_{coll}}{2} + \frac{s_0 \cdot t'_{slot}}{p_{idle\_cp}} \right) + \sum_{k=1}^{\ell}\left(b_k \cdot \frac{s_k}{p_{idle\_cp}} \right) \cdot t'_{slot}
\end{aligned} \tag{4}
$$

The next step is to compute $b_k = f(n, r)$.

2.3. Backoff Stage Probabilities

During the vulnerable period there are only two possible events: success and collision. It is reasonable to assume that the probabilities of these events, p_{succ_vp} and p_{coll_vp} respectively, are independent of the node's current backoff stage. Then, from a packet's perspective, we can represent the backoff-stage counting process K as a Markov chain in Fig. 4. We can derive b_k is from the stationary distribution for the Markov chain, $b_k = \Sigma\, [b_j \cdot \Pr(K = k \mid K = j)]$. For example,

$$b_0 = b_0 \cdot p_{succ\_vp} + b_1 \cdot p_{succ\_vp} + \ldots + b_\ell \cdot 1$$

and generally

$$b_k = (p_{coll\_vp})^k \cdot b_0, \qquad k = 1, 2, \ldots, \ell$$

We do not consider the "drop" state and b_{drop} because there is no attempt after the packet is dropped. Only the sates 0, …, ℓ count since in these a transmission will be attempted at the end of the inter-event period.

We can write this in a matrix form $\mathbf{b} = \mathbf{A} \cdot \mathbf{b}$, where $\mathbf{A}$ is the transition probability matrix in steady state. Given the constraint that $p_{succ\_vp} + p_{coll\_vp} = 1$ and $\Sigma\, b_k = 1$, we can solve for $\mathbf{b}$. Thus, $\mathbf{b}$ is the eigenvector of the transition matrix with eigenvalue 1:

$$
\begin{bmatrix}
p_{succ\_vp} - 1 & p_{succ\_vp} & p_{succ\_vp} & \cdots & p_{succ\_vp} & 1 \\
p_{coll\_vp} & -1 & 0 & \cdots & 0 & 0 \\
0 & p_{coll\_vp} & -1 & \cdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & p_{coll\_vp} & -1 & 0 \\
0 & 0 & \cdots & 0 & p_{coll\_vp} & -1
\end{bmatrix}
\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ \cdots \\ b_{\ell-1} \\ b_\ell \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ 0 \\ 0 \end{bmatrix} \tag{5}
$$

The probabilities of success and collision, p_{succ_vp} and p_{coll_vp} respectively, are derived above in Eqs. (2), and these two equations combined will yield b_k. The probability of packet being dropped will be considered in Section 4.1 below.

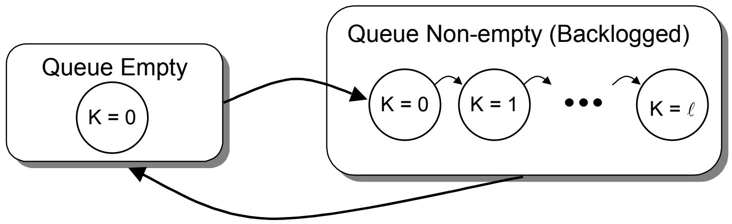

Fig. 5. Only two states of the transmission queue are distinguished: Empty and Backlogged. The station can be in different backoff stages (k = 0, 1, 2, 3, …, ℓ) only when backlogged.

Notice that we are ignoring the queuing at individual nodes as well as the application traffic profile λ_{Appl}. In other words, we ignore the length of the queue and just consider two states: whether the queue is empty or non-empty (Fig. 5). For example, we assume a real-time traffic with bounded delay per packet, so the stale packets get purged from the queue. This prevents queue buildup and so n is not function of the collision probability.

2.4. Putting It All Together

Thus far we derived the expected rate r of frame transmissions for a single node. After plugging Eq. (4) into (3), we obtain:

$$r = \frac{1}{t_{ep}} = \frac{1}{DIFS + b_0 \cdot \left(\frac{p_{succ\_vp} \cdot t_{succ} + p_{coll\_vp} \cdot t_{coll}}{2} + \frac{s_0 \cdot t'_{slot}}{p_{idle\_cp}} \right) + \sum_{k=1}^{\ell} \left(b_k \cdot \frac{s_k}{p_{idle\_cp}} \right) \cdot t'_{slot}} \tag{6}$$

The probabilities q_c and q_v used in Eqs. (1) and (2) are given as:

$$q_c = r \cdot t_{slot}$$

$$q_v = r \cdot t_{vp}$$

Therefore, we obtain $r = fct(n, r)$ and for each n, there exists a corresponding r that satisfies (6). This nonlinear equation does not have a closed form solution. It can be solved numerically, e.g., using the Newton's iteration method, with n as the independent variable. If we set $y = fct(n, r) - r = 0$, then Matlab function `fzero()` can be used to determine the solution.

Plot here a diagram of r vs. n.

After obtaining numerically individual points of $r = fct(n)$, we can try curve fitting to obtain an analytical approximation for $r = fct(n)$.

2.5. Vulnerability Period

Unlike other methods for 802.11 performance analysis, our method specifies the collision probability for the actual vulnerability period. What we can do here is to recover $r = fct(n)$ under different definitions of t_{vp}. For example, we could solve Eq. (6) when t_{vp} includes SIFS after RTS because of hidden stations.

Perform here rate analysis with different definitions of the vulnerability period.

3. Delay Analysis

We compute the expected delay per packet as a sum of expected delays at each backoff stage. At each stage, an average node first suffers preemptions from all the nodes with the backoff countdown smaller than its own. Then it attempts transmission. If collision occurs, the node increments the backoff stage and starts anew.

The expected waiting time at a k^{th} backoff stage until an attempted transmission was derived above as:

$$t_{start}(k) = \text{DIFS} + \begin{cases} \dfrac{p_{succ\_vp} \cdot t_{succ} + p_{coll\_vp} \cdot t_{coll}}{2} + \dfrac{s_0 \cdot t'_{slot}}{p_{idle\_cp}}, & k = 0 \\ \dfrac{s_k \cdot t'_{slot}}{p_{idle\_cp}}, & k = 1,2,\ldots,\ell \end{cases}$$

The expected delay per packet is:

$$W = \sum_{k=0}^{\ell}\left(b_k \cdot \sum_{j=0}^{k} t_{start}(j)\right) = \sum_{k=0}^{\ell}\left[b_k \cdot \left(\frac{p_{succ\_vp} \cdot t_{succ} + p_{coll\_vp} \cdot t_{coll}}{2} + \frac{s_0 \cdot t'_{slot}}{p_{idle\_cp}} + \sum_{j=1}^{k}\frac{s_k \cdot t'_{slot}}{p_{idle\_cp}}\right)\right] \quad (7)$$

This procedure is somewhat different from the delay derivation presented in [Carvalho & Garcia-Luna-Aceves 2003]. Is it possible to establish equivalency?

Plot here the diagram of W vs. p_{coll} or vs. n and other diagrams from [Carvalho & Garcia-Luna-Aceves 2003].

4. Throughput Analysis

4.1. Dropped Packets

In the above analysis of an average packet delay, we do not consider the packets that are discarded after the retry limit ℓ is reached. However, these discarded packets waste the channel capacity and in turn reduce the throughput of the system. For a discarded packet, the channel time it wastes is:

$$t_{drop} = \sum_{k=0}^{\ell} t_{coll,k}$$

with the probability

$$p_{drop} = 1 - \sum_{k=0}^{\ell} b_k = 1 - \sum_{k=0}^{\ell}\left(\left(p_{coll\_cp}\right)^k \cdot p_{succ\_cp}\right) = 1 - p_{succ\_cp} \cdot \frac{p_{coll\_cp} \cdot \left(1 - p_{coll\_cp}^{\ell}\right)}{1 - p_{coll\_cp}}$$

4.2. Channel Throughput

The packet service time starts when the packet becomes the head of its queue and ends when its positive acknowledgement is received. This is on the average $W + t_{xmit}$. During the average packet delay period W, there will be on the average this many preemptive transmissions:

$$n_i = \sum_{k=0}^{\ell} b_k \cdot s_k \cdot \frac{1 - p_{idle\_cp}}{p_{idle\_cp}} \stackrel{?}{=} (n-1) \cdot r \cdot t_{ep}$$

Of these, ($p_{succ\_cp} \cdot n_i$) will be successfully completed and the rest will end up in collision. The dropped packets count as "idle" channel time. Therefore, during the average service time there will be ($p_{succ\_cp} \cdot n_i$) + 1 packets served and the throughput is:

$$\gamma = \frac{(1 - p_{drop}) \cdot E[P]}{p_{drop} \cdot t_{drop} + (1 - p_{drop}) \cdot (W + t_{xmit})} \cdot (p_{succ\_cp} \cdot n_i + 1)$$

4.3. Number of Backlogged Stations

So far we assumed that the number of backlogged stations n is constant and unknown. We could estimate it as follows. We have m Poisson inputs so the aggregate input rate is $m \cdot \lambda_{Appl}$, and the aggregate is also a Poisson process. So, we can consider the whole system as a $M/G/1$ queue with a single server, where the channel is the server. All we need is to figure out what is the service rate of this server. The packet service time is $t_{svc} = W + t_{xmit}$, i.e., the average delay time plus the packet transmission time (for an average packet length). Then the channel service rate is:

$$\mu = 1 / t_{svc} \tag{8}$$

From here, we can just use the Pollaczek-Khinchin formula [Bertsekas & Gallagher 1992] and determine N_Q, the average number of packets in the queue. The P-K formula is valid if there exist the steady-state averages W and N_Q. So it is required that $m \cdot \lambda_{Appl} \leq \lambda_c$; otherwise, as the time goes to infinity, W and N_Q will grow to infinity.

Since all m stations are identical, it is safe to state that all of their queues are statistically identical. With small N_Q (i.e., less than m) we assume that it is N_Q backlogged stations with one packet each and the remaining $m - N_Q$ stations are idle. With large N_Q, all m stations are backlogged since they are statistically identical. Therefore,

$$n = \begin{cases} N_Q, & N_Q < m \\ m, & N_Q \geq m \end{cases}$$

For a steady state, the number n of backlogged nodes is constant, although individual nodes may switch between backlogged and idle states. Then, overall arrival rate must be equal to the departure rate (or else there will be an infinite queue buildup):

$$m \cdot \lambda_{Appl} = \gamma$$

This nonlinear equation in n must be solved numerically. One trouble here may be that service time depends on n which depends on the arrival rate?

5. Summary

We summarize the above model as follows:

$$n \leftrightarrow b_k \leftrightarrow r \leftrightarrow \lambda_c \leftrightarrow [p_{succ}, p_{coll}] \leftrightarrow [\gamma, W]$$

where γ symbolizes the expected channel throughput.

There is a tradeoff to consider here between the fairness and link utilization, which can be optimized by varying the frame length.

References

1. F. Alizadeh-Shabdiz and S. Subramaniam, "A finite load analytical model for the IEEE 802.11 distributed coordinated function MAC," *Proceedings of the Workshop on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt'03)*, Sophia-Antipolis, France, pp. 321-322, March 2003.
2. D. Bertsekas and R. Gallagher, *Data Networks*. 2nd edition, Prentice Hall, Upper Saddle River, NJ, 1992.
3. G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535-547, March 2000.
4. M. M. Carvalho and J. J. Garcia-Luna-Aceves, "Delay analysis of IEEE 802.11 in single-hop networks," *Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03)*, Atlanta, GA, pp. 146-155, November 2003.
5. P. Chatzimisios, A. C. Boucouvalas, and V. Vistas, "Packet delay analysis of IEEE 802.11 MAC protocol," *IEE Electronics Letters*, vol. 39, no. 18, pp.1358-1359, 4th September 2003.
6. C. H. Foh and M. Zukerman, "A new technique for performance evaluation of random access protocols," *Proceedings of the IEEE International Conference on Communications (ICC 2002)*, vol. 4, pp. 2284-2288, New York, NY, April 2002.
7. C. H. Foh and M. Zukerman, "Performance analysis of the IEEE 802.11 MAC protocol," *Proceedings of European Wireless 2002—Next Generation Wireless Networks: Technologies, Protocols, Services and Applications*, Florence, Italy, February 2002. Online at: http://www.ntu.edu.sg/home/aschfoh/pub/publications.html
8. J. Jun, P. Peddabachagari, and M. Sichitiu, "Theoretical maximum throughput of IEEE 802.11 and its applications," *Proceedings of the 2nd IEEE International Symposium on Network Computing and Applications (NCA'03)*, Cambridge, MA, pp. 249-256, April 2003.
9. H. Wu, Y. Peng, K. Long, S. Cheng, and J. Ma, "Performance of reliable transport protocol over IEEE 802.11 wireless LAN: Analysis and enhancement," *Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2002)*, vol. 2, pp. 599-607, June 2002.
10. E. Ziouva and T. Antonakopoulos, "CSMA/CA performance under high traffic conditions: Throughput and delay analysis," *Computer Communications (Elsevier)*, vol. 25, no. 3, pp.313-321, 15 February 2002.