Throughput analysis and protocol design for CSMA and BTMA protocols under noisy environments

J.-H. Huang
L. Kleinrock

Indexing terms: Radiocommunication, Packet radio network

Abstract: In using a CSMA (Carrier-Sense Multi-Access) protocol in a packet radio network, we cannot ignore the probability that the environment is so noisy that the channel state is misdetected. Similarly, with a BTMA (Busy-Tone Multi-Access) protocol, we must take into account the probability of misdetecting the busy or idle state of the channel. The paper focuses on these problems and tries to evaluate the throughput performance in p-persistent access schemes, using an iterative approximation method. Furthermore, a new protocol is suggested, which can effectively cope with the imperfect sensing property to achieve an improved throughput by sensing the channel in several consecutive slots.

1 Introduction

As with the recent proliferation of mobile cellular phones, computer traffic through the wireless sky is attracting more attention. However, since computer traffic is much more bursty than the traffic of a telephone conversation, the channel allocation scheme used by mobile cellular phones may not be adequate for computers. Under such a circumstance, a multiaccess scheme may be more attractive for computer communication. Since the early 1970s, there have been many multiaccess protocols [1–8] proposed in the literature for wireless communication. Among these, CSMA and BTMA are good candidates for packet radio networks. In this paper, we consider the p-persistent access schemes under CSMA and BTMA protocols which are applied in a noisy environment, such that the channel state may be misdetected.

A performance analysis of CSMA and BTMA can be found in Reference 8. Here, Tobagi analysed the performance of CSMA using a p-persistent access scheme, assuming that the detection of the channel carrier will always be correct. In the same work, Tobagi analysed the performance of BTMA using a non-persistent protocol, assuming that the detection of the channel is not perfect. In this paper, we consider CSMA and BTMA using p-persistent access schemes. The performance degradation when the sensor makes mistakes is presented. This paper shows that the throughput is very sensitive to the correct sensing of a busy channel. Based on this observation, we propose a new protocol that improves the throughput performance under noisy environments.

Renewal theory is used to find the throughput performance. We define the *idle period* to be the time during which the channel is idle and no terminals are ready for transmission. An *initial delay period* is the time during which the channel is idle but there is at least one ready terminal trying to access the channel. A *transmission period* is the time during which the channel is busy transmitting. A packet transmission is successful only if there is no collision during its transmission time. From the above definition, we define a *cycle* to be the interval from the beginning of one idle period until the beginning of the next. Within a cycle, there is one *idle period*, several *initial delay periods* and several *transmission periods*, as shown in Fig. 1.

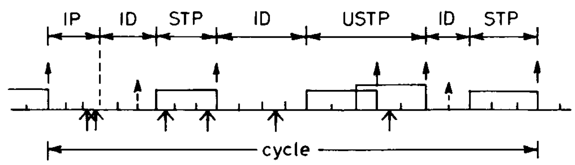

Fig. 1 *An example of a cycle, showing idle periods, initial delay periods and transmission periods*

↑ arrival
departure with transmission
departure without transmission
IP: idle period
ID: initial delay period
STP: successful transmission period
USTP: unsuccessful transmission period

Note that a packet may be rescheduled for retransmission without being transmitted when the channel is idle, as shown in Fig. 1. This is caused by a busy station, which first detects the idle channel but does not transmit, then misdetects the channel as being busy (false alarm). The details of the protocol are provided in Section 2. We define the normalised throughput to be the expected fraction of the successful transmission time during a cycle.

It is too complicated to obtain an exact solution for such a system environment. This is because the length of the ith initial delay period depends upon the number of packets remaining from the $(i-1)$th transmission period, which depends on the number of packets remaining from the $(i-1)$th initial delay period, which again depends upon the length of the $(i-1)$th initial delay period, etc. It is this coupling phenomenon that makes the analysis difficult. An iterative approximation method is therefore used to find the throughput performance, and some simulation results are used to verify the approximation. The approximation results are quite close to the simulation

Paper 8779I (E7, E8), first received 4th March 1991 and in revised form 15th January 1992

J.-H. Huang is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, Republic of China

L. Kleinrock is with the Department of Computer Science, University of California, Los Angeles, Los Angeles, California, USA

results for various system parameter values. Furthermore, we also identify some features which are important in implementation.

2 Channel model and the system characteristics

The model used for CSMA in this paper is a fully connected packet radio network, and the traffic is between terminal pairs. For BTMA, the model used is a hidden-terminal model, that is, there is a central station with many terminals surrounding it. The central station can hear all the terminals and all the terminals can hear the central station, but the terminals cannot hear each other. Each terminal will try to transmit a packet to the central station if there is one in its buffer. The central station receives packets from the terminals but transmits none back to the terminals. However, it will transmit a busy tone on the busy-tone channel whenever it is receiving signals from the data channel. In this paper, we ignore the bandwidth required to transmit the busy tone; we assume that the entire channel bandwidth is for normal message transmission only.

For CSMA and BTMA, we assume that each terminal has only one buffer; hence there will be no queuing at a terminal. We also assume that there is an infinite number of terminals. The arrival process to the set of terminals is a Poisson process, and every packet has a fixed length. The access prototol we use is p-persistent. The following is a review of this protocol [9]. Time is allocated to fixed-duration slots and all ready terminals are required to begin their transmission only at the beginning of a slot. A packet transmission takes N slots.

Consider a ready terminal: if this terminal senses the channel to be idle, then with probability p, the terminal transmits the packet. With probability $1 - p$, the terminal delays the packet transmission by one slot. If at this new point in time the channel is still detected as being idle, the above process is repeated; otherwise, a packet must have started transmission, and our terminal schedules the retransmission of its packet according to the retransmission delay distribution.

If the ready terminal senses the channel as being busy, it waits until it becomes idle (at the end of the current transmission) and then operates as above.

Based on the model description, the analysis of CSMA will be the same as that of BTMA. Hence, everything described below can be applied to both protocols. As we do not assume that the detection will be always correct, we define two parameters to specify this imperfect sensing:

D = probability of correctly detecting a signal transmission

F = probability of falsely detecting a signal transmission when there is no transmission (i.e. a false alarm)

Obviously, the signal mentioned above is the busy tone in BTMA, and is the packet transmission signal itself in CSMA. There are a number of design parameters in this model. We define G to be the total arrival rate during one packet transmission time, and assume that the total arrival process is Poisson. Hence, all the design parameters of this model are D, F, N, p, and G. Note that there is a constraint on choosing N: we cannot choose N to be arbitrarily large, since for CSMA the slot size cannot be smaller than the one-way propagation delay. For BTMA, the slot size cannot be smaller than twice the propagation delay, as we need one propagation delay for the packet transmission from the terminal to the central station, and one propagation delay for the busy-tone transmission from the central station back to the terminals. For most cases, N has a value between 20 and 100.

3 Model assumptions and analysis

In this section, we describe the assumptions made in the approximation and the reasons to support those assumptions. From those assumptions, we are able to use an iterative scheme to obtain the system throughput. The detailed throughput analysis is provided in the Appendix.

3.1 Model assumptions

As the coupling of the initial delay periods and the transmission periods mentioned in Section 1 makes the analysis difficult, we break up this coupling by the following two assumptions. We assume that a cycle consists of one idle period, L initial delay periods and L transmission periods (L is a parameter to be computed). In order to find the throughput, we must find the average length of the idle period, the initial delay periods and the transmission periods, and the fraction of successful transmission periods among all transmission periods. Although the average length of the idle period can be found easily, the average lengths of the initial delay and transmission periods are not.

The first assumption we make is that all transmission periods and initial delay periods have the same length distribution, except the first one. The reason we make the first initial delay period different is because it is started by the packets which arrived in the last slot of the idle period, whereas the other initial delay periods are started by those ready terminals accumulated at the end of the preceding transmission period. Similarly, the reason we make the first transmission period different is because it is started by those ready terminals accumulated at the end of the first initial delay period, whereas the other transmission periods are started by initial delay periods which are not the first.

Secondly, we assume that a packet transmission can be hit once at most, and the hit will occur right on the tail of the packet transmission. This is a bold assumption, but we will show why it is reasonable. Let us first define A to be the probability that a transmission period is successful, that is, a packet transmission is not hit (the hit would have resulted in a collision) during the entire transmission. If the value of A is high, say $A \geqslant 0.8$, then the probability of a transmission being hit more than once would be

$$1 - \text{Prob[transmission is not hit]}$$
$$- \text{Prob[transmission is hit exactly once]}$$

which equals $1 - A - A(1 - A) = (1 - A)^2$. This probability will be smaller than 0.04 for this range of A values. Thus, neglecting multiple hits is a reasonable assumption when the value of A is high.

The other assumption, that of hitting right on the tail, is justified as follows. As arrivals during a packet transmission period are uniformly distributed, the number of accumulated ready terminals which have the potential to hit the transmission will keep increasing during the transmission period, with respect to time. Hence, the probability of the transmission being hit by these ready terminals misdetecting the channel state is also an increasing function with respect to time. Therefore, if this transmission does get hit, it will tend to be hit closer to the tail of the transmission period.

Furthermore, there is a mutual compensation for these two assumptions, which motivates us to make the two assumptions together. By assuming there can be one hit at most during a transmission period, we underestimate the length of the collided transmission period. However, by assuming that the hit will happen exactly on the tail, we overestimate the length of the collided transmission period. Therefore, they tend to compensate each other to reduce the error. More surprisingly, when the value of A is as small as 0.2 or 0.3, the results obtained from this approximation are also fairly close to the simulation results, although not as good as with a large value of A. This is also due to the mutual compensation effect.

Concerning a practical implementation issue, we would always like to make the value of A high. The reason is obvious, as we do not want the probability of collision to be high because it will degrade the throughput performance as well as the delay performance. By making these assumptions, and by finding the value of A, we find that the average length of a transmission period is $[N \times A + 2N(1 - A)]$ slots long, which equals $(2 - A)N$ slots long. Hence, the remaining task is to find the average length of the initial delay period and the fraction of successful transmission periods.

3.2 Flow of the analysis

To find the average length of the initial delay periods takes more effort. The tool used here is a recursive scheme using the following notation:

MID = the mean length of an initial delay period
NBI = number of ready terminals at the beginning of an initial delay period
NBT = number of ready terminals at the beginning of a transmission period

In the following, we explain how the iteration works. In order to find the MID of the ith initial delay period, we must find the NBI of the ith initial delay period. However, in order to find the NBI of the ith initial delay period, we have to find the NBT of the $(i - 1)$th transmission period. Again, in order to find the NBT of the $(i - 1)$th transmission period, we have to know the MID of the $(i - 1)$th initial delay period and the NBI of the $(i - 1)$th initial delay period. Here we see the recursion.

The strategy used here is to give an arbitrary value to the NBI, then to proceed as described above to find the MID of the initial delay period, and then the NBT. From this NBT, we calculate the value of the NBI accordingly. If this calculated NBI is different from the arbitrarily assigned value of the NBI, we then repeat this process using the calculated NBI to find another new calculated NBI. This process is repeated until the NBI is stabilised. By actually running more than 1000 sets of data, we found that only two iterations were required to stabilise these values. There is a good reason for this, which will be shown during the derivation.

The flow of the algorithm is as follows:

(i) Calculate the average length of an idle period, denoted as IP. In Reference 8, it is shown that

$$IP = \frac{1}{1 - e^{-G/N}}$$

(ii) Let NBI_0 be the NBI of the first initial delay period. Note that NBI_0 is the number of packets that arrive in the last slot of the idle period. Hence, NBI_0 equals

$$\frac{G/N}{1 - e^{-G/G}}$$

We then find the expected length of the NBT of the first initial delay period (denoted as MID_0 and NBT_0) from this NBI_0.

(iii) From NBT_0 we find A_0, which is the probability that the first transmission period is a successful one, and NBI_1 for the first transmission period. Then we use this NBI_1 to be the initial NBI value to start the recursion.

(iv) Use NBI_1 to find the MID and the NBT. Use this NBT to find a new set of A and NBI. Then compare this NBI with NBI_1. If the difference is greater than ε (ε is a precision parameter set by the user), we use the new NBI to repeat this step until the NBI converges (A, NBT and MID will also converge when the NBI converges).

(v) Having A, we can find the average length of the transmission period as $[N \times A + 2N(1 - A)] = (2 - A)N$ slots, by the approach described earlier.

(vi) The last step of this algorithm is to find the average value of L, which can be found easily as shown in the Appendix. Having all these, we finally find the throughput S, applying renewal theory as follows:

$$S = \frac{A_0 \times N + (L - 1) \times A \times N}{IP + MID_0 + (L - 1) \times MID + (2 - A_0)N + (L - 1) \times (2 - A)N} \quad (1)$$

The details of the recursive analysis are provided in the Appendix. In the next Section, some simulation results are presented and discussed.

4 Simulation results and discussions

In this section we will compare the results derived both from the analysis and from the simulations. For each comparison we will change only one parameter and keep the others unchanged to learn the effect of the changing parameter. In Figs. 2–11, the solid line represents the analytical results and the dots represent the simulation results.

The first comparison is made by changing the value of N, as shown in Figs. 2, 3, and 4. Note that Fig. 4 is the

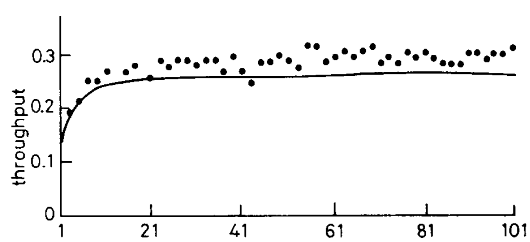

Fig. 2 *Throughput versus N (D = 0.7, F = 0.1, p = 0.01, G = 5)*

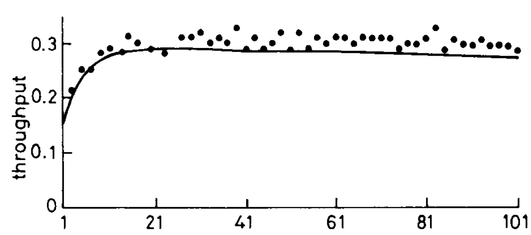

Fig. 3 *Throughput versus N (D = 0.9, F = 0.1, p = 0.01, G = 5)*

case when the detection is perfect, i.e. $D = 1$ and $F = 0$. It is shown in Fig. 4 that when the detection is perfect, the throughput will keep increasing as N increases. Hence, we should choose the slot size to be the propagation delay, in order to have the largest value of N. However,

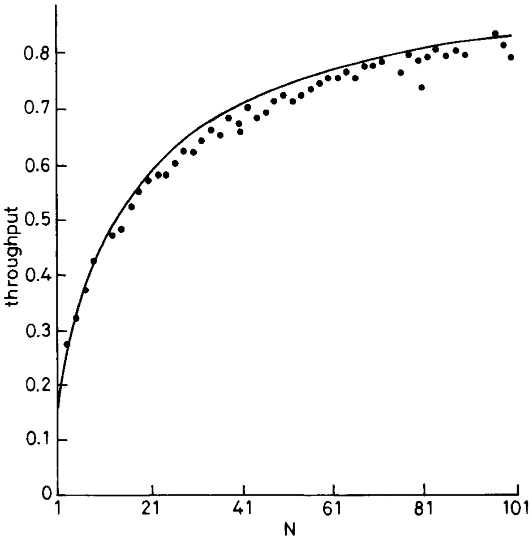

Fig. 4 *Throughput versus N (D = 1, F = 0, p = 0.01, G = 5)*

from Figs. 2 and 3 we see that this is not the case when the detection is not perfect. Here, some tradeoffs can be seen. Larger N means a smaller slot size; hence, we can shorten the idle period and the initial delay period, which is good. However, a large N introduces an increased probability that a transmission will be hit (that is, $1 - A$ increases), which is bad. From Figs. 2 and 3 we see that both the analysis and the simulation indicate that the throughput will increase sharply as N starts increasing from 1, and then remain almost unchanged over a wide range of N (e.g. $N > 20$). This result suggests that we should choose N to have a moderate value, so that the throughput curve starts to level off. By so doing, we have a wider slot size and more time to detect the channel state, which will increase the value of D and decrease the value of F, which can further improve the throughput.

The second comparison is made by changing the value of p, as shown in Figs. 5 and 6. From Fig. 6, we see that

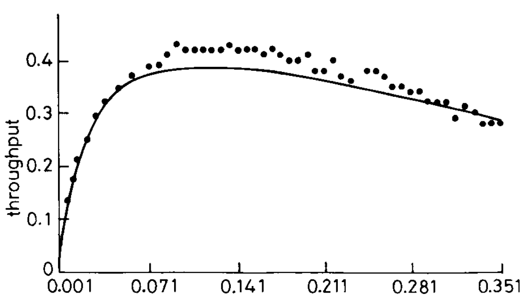

Fig. 5 *Throughput versus p (N = 50, D = 0.9, F = 0.1, G = 2)*

the throughput is very sensitive to a change in p around the peak when the traffic is high, but it is not so sensitive when the traffic is low, as shown in Fig. 5. Therefore, we must choose the value of p carefully, according to the system traffic.

The third comparison is made by changing the value of D, as shown in Figs. 7, 8, 9, and 10, and the last comparison is made by changing the value of F, as shown in Figs. 11, 12, 13, and 14. We put these two sets of comparisons together because there is a relation between D and F. In Reference 10, it is shown that D is actually a function of F:

$$D = F^{1/1+\mu} \quad (2)$$

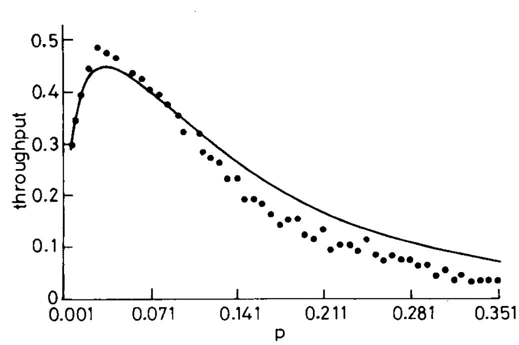

Fig. 6 *Throughput versus p (N = 50, D = 0.9, F = 0.1, G = 6)*

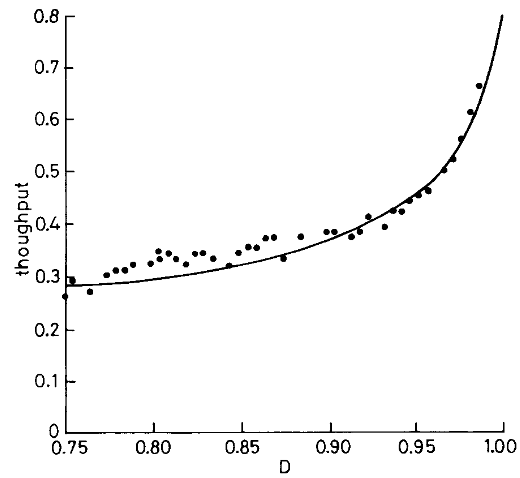

Fig. 7 *Throughput versus D (N = 50, F = 0.1, p = 0.1, G = 5)*

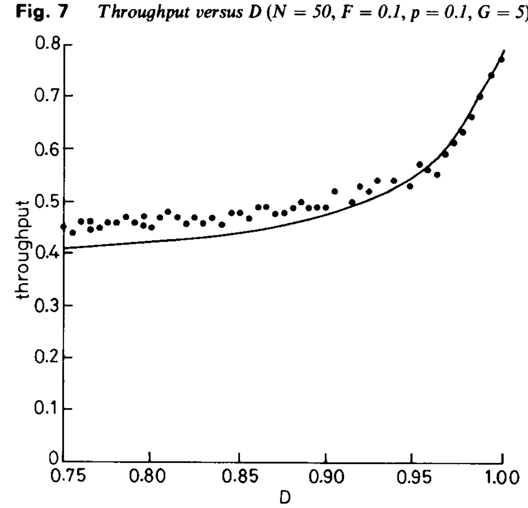

Fig. 8 *Throughput versus D (N = 50, F = 0.01, p = 0.1, G = 2)*

In eqn. 2, μ is the signal-to-noise ratio when the busy tone is present over the entire detection time (also defined as a window) on the busy-tone channel. The signal-to-noise ratio is a function of the bandwidth of the busy-tone channel. This equation shows that we can increase F to increase D.

We also arrange the values of p, G and N to be the same for Figs. 7 and 11, so that we can make some comparisons. The same arrangements are made between Figs. 8 and 12, Figs. 9 and 13, and Figs. 10 and 14.

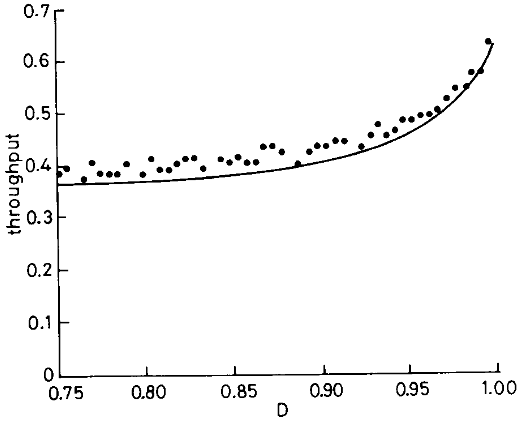

Fig. 9 *Throughput versus D (N = 50, F = 0.1, p = 0.01, G = 10)*

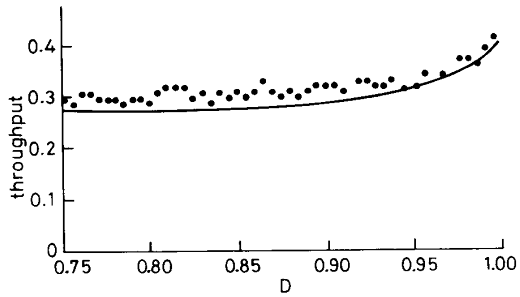

Fig. 10 *Throughput versus D (N = 50, F = 0.1, p = 0.01, G = 5)*

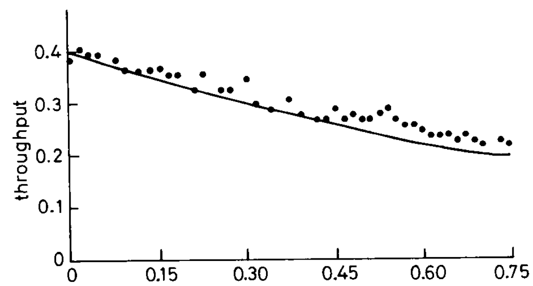

Fig. 11 *Throughput versus F (N = 50, D = 0.9, p = 0.1, G = 5)*

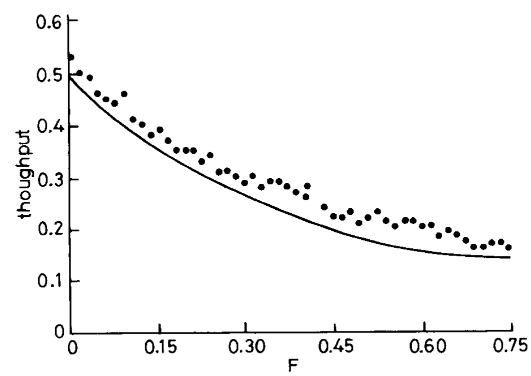

Fig. 12 *Throughput versus F (N = 50, D = 0.9, p = 0.1, G = 2)*

Note that the throughput deteriorates significantly when D is slightly deviated from 1 when other parameters are fixed, and this behaviour is shown particularly in Fig. 7. Also note that the throughput is rather insensitive to F, as shown in Fig. 11. Figs. 7 and 11 suggest that we can increase F to increase D, to improve the throughput in the case where $p = 0.1$, $G = 5$ and $N = 50$. However, in

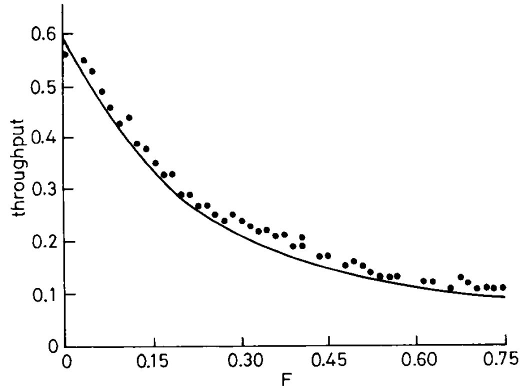

Fig. 13 *Throughput versus F (N = 50, D = 0.9, p = 0.01, G = 10)*

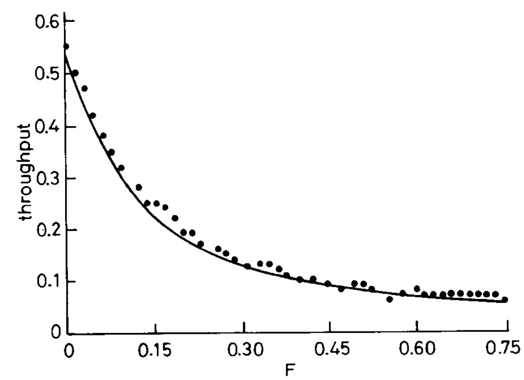

Fig. 14 *Throughput versus F (N = 50, D = 0.9, p = 0.01, G = 5)*

Fig. 12 it is shown that the throughput is also sensitive to F. Hence, we cannot simply increase F to increase D and hope for a throughput improvement in the case where $p = 0.1$, $G = 2$ and $N = 50$. Moreover, Figs. 10 and 14 illustrate a totally different behaviour. In these two figures, it is shown that the throughput is insensitive to D but very sensitive to F in the case where $p = 0.01$, $G = 5$ and $N = 50$.

The reason for this is that if the value of F is much higher than the value of p, then most of the arriving packets will be rescheduled for retransmission before being transmitted, since they detect a false alarm before they get a chance to transmit. This effect will make the initial delay period very long, and degrade the throughput. If we increase the value of F under this circumstance (as shown in Fig. 14), this effect would be even stronger and decrease the throughput dramatically.

Observing from Figs. 7–14, we suggest a useful design rule-of-thumb: when the product of G and p is high (say, >0.5) then the throughput is very sensitive to D and insensitive to F. This suggests that we should choose a higher value of F to achieve a subsequent higher value of D, to achieve a better throughput. On the contrary, if the product of G and p is low (say, <0.05), then the throughput is sensitive to F and insensitive to D. This suggests that we should choose a lower value of F, even if it would induce a lower value of D, to achieve a better throughput. If the product of G and p is neither high nor low (say, between 0.05 and 0.5) then the throughput is sensitive to both D and F. In this case we offer no rule of thumb in choosing the values of D and F.

Usually, we prefer to have a higher value of A to obtain a higher throughput. Also, we know that the value of A depends heavily on the value of D. It is shown in eqn. 2 that we can increase the value of D by increasing the value of F, but the discussion above shows that we cannot simply increase the value of F to increase the value of D (and hence increase the value of A and the throughput) without considering the negative effect on throughput when F is increased.

5 A new protocol

From the preceeding Sections we know that the throughput performance is poor when the value of D, and hence A, is low. The reason for this is that when the value of A is low, the packet transmission is easily hit by those terminals which, due to errors, sense the channel state to be idle. Each collision costs at least N slots, and these costly collisions reduce the throughput.

In order to improve the throughput, we would like to increase the value of A. In the preceding Sections we showed several ways of increasing the value of A. One way is by decreasing the value of N, but from Figs. 2, 3 and 4 we can see that it might be costly when the value of N is lower than a critical value.

Another way is to increase the value of F, increasing the value of D and hence increasing the value of A. Again, we have shown that under some circumstances, it would also be costly by increasing the value of F. Figs. 13 and 14 are some examples.

Here, we propose a new protocol which can increase the value of A without affecting any other parameters. This new protocol is only slightly different from the original one, in that in the new protocol, every ready terminal will transmit only after a fixed number of consecutive idle states (NOI) is detected. By so doing, we reduce the probability of an ongoing transmission being hit, hence increasing the value of A. The throughput analysis of the new protocol is basically identical to the one we already derived.

Fig. 15 shows the improvement of this protocol over the original protocol by changing G with NOI equal to 1,

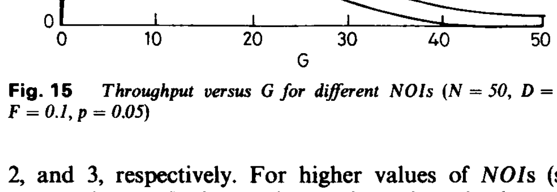

Fig. 15 *Throughput versus G for different NOIs ($N = 50$, $D = 0.9$, $F = 0.1$, $p = 0.05$)*

2, and 3, respectively. For higher values of NOIs (say $NOI = 4$, 5 or 6), the maximum throughput is about the same as that achieved with $NOI = 3$. The reason for this is that when $NOI = 3$, the value of A is very close to 1, already at the optimal point. Increasing the value of NOI, therefore, will not improve the throughput any further.

This protocol also incurs a negative effect. When the channel state is idle, each ready terminals needs at least $NOI - 1$ more slots before transmission than the original protocol. This makes the initial delay period longer and tends to make the throughput poorer. Fig. 15 shows this phenomenon when $NOI = 15$. However, if we choose the value of NOI to be much smaller than the value of N, then the positive effect predominates over the negative effect and will enhance the throughput.

6 Conclusions

This paper investigates a method to find the throughput of a single-hop packet radio network with imperfect sensing for certain p-persistent protocols, a real-life situation. The reason for the discrepancies between the analytical results and the simulation results comes from the assumption of 'hitting right on the tail' and of 'omitting multiple hits'. However, the analytical method has successfully captured some important characteristics of the system behaviour (e.g. the sharp rise of the curves in Figs. 6 and 7 as D approaches unity) and has also achieved results reasonably close to the simulation results. From these results, we propose some rules-of-thumb in choosing the system parameters to obtain a better performance. A new protocol is proposed for this, which is intuitively reasonable and indeed has better performance, as shown by the simulations.

7 Acknowledgment

This work was supported by the Defense Advanced Research Projects Agency under Contract MDA 903-87-C0663, Parallel Systems Laboratory.

8 References

1 ABRAMSON, N.: 'Development of the ALOHANET', *IEEE Trans. Inform. Theory*, March 1985, **IT-31**, pp. 119–123
2 KLEINROCK, L., and TOBAGI, F.: 'Random access techniques for data transmission over packet-switched radio channels', *Proc. NCC*, 1975, pp. 187–201
3 CHLAMTAC, I., and KUTTEN, S.: 'On broadcasting in packet radio–problem analysis and protocol design', *IEEE Trans. Commun.*, Dec. 1985, **COM-35**, pp. 1240–1246
4 JUBIN, J.: 'Current packet radio network protocols', *Proc. INFOCOM*, 1985
5 TOBAGI, F., BINDER, R., and LEINER, B.: 'Packet radio and satellite networks', *IEEE Commun. Magazine*, Nov. 1984, **22**, pp. 24–40
6 CAPETANAKIS, J.I.: 'Tree algorithms for packet broadcast channels', *IEEE Trans. Inform. Theory*, Sept. 1979, **IT-25**, pp. 505–515
7 MOLLE, M.L.: 'Unifications and extensions of the multiple access communications problem'. CSD Report No. 810730 (UCLA-ENG-8118), Computer Science Department, University of California, Los Angeles, July 1981
8 TOBAGI, F.A.: 'Random access techniques for data transmission over packet switched radio networks'. UCLA-ENG-7499, Computer Science Department, University of California, Los Angeles. PhD dissertation, December 1974
9 KLEINROCK, L.: 'Queueing systems: Vol. II — Computer applications' (John Wiley & Sons, New York, 1976)
10 WAINSTEIN, L.A., and ZUBAKOV, V.D.: 'Extraction of signals from noise' (Prentice Hall, Englewood Cliffs, New Jersey, 1962)

9 Appendix: Recursive analysis of the throughput

To obtain the throughput expression, we have to find the values of A, NBI, MID, NBT and L. The following notation is defined:

$HB(m, k)$ is the probability of a transmission being hit in any slot from k to m by a packet arriving in the kth slot of that transmission, given that the length of the transmission period equals m slots.

$HI(k)$ is the probability that a packet does not transmit for at least $(k > 0)$ slots after its arrival, given the channel is actually idle.

$S(m, k)$ is the probability that a packet which arrives at the kth slot of a transmission period, stays without leaving (scheduled for retransmission before transmitting) or transmitting during the transmission, given that the length of the transmission period is m slots long.

From these definitions, we have

$$HB(m, k) = \bar{D}p + (D\bar{D}p + \bar{D}q\bar{D}p) + \cdots + (D^{m-k}\bar{D}p + D^{m-k-1}\bar{D}q\bar{D}p + \cdots + \bar{D}^{m-k}q^{m-k}\bar{D}p)$$
$$= \frac{\bar{D}^2 pq}{\bar{D}q - D} \times \left[1 - \frac{(\bar{D}q)^{m-k+1}}{1 - \bar{D}q} - \frac{D}{\bar{D}q}\frac{1 - D^{m-k+1}}{\bar{D}}\right] \quad (3)$$

$$HI(k) = \bar{F}p + (F\bar{F}p + \bar{F}q\bar{F}p) + \cdots + (F^{k-1}\bar{F}p + \cdots + \bar{F}^{k-1}q^{k-1}\bar{F}p)$$
$$= 1 - \frac{\bar{F}^2 pq}{\bar{F}q - F} \times \frac{1 - (\bar{F}q)^k}{1 - \bar{F}q - \frac{F}{\bar{F}q}\frac{1 - F^k}{\bar{F}}}$$

$$S(m, k) = D^{m-k+1} + D^{m-k}\bar{D}q + D^{m-k-1}(\bar{D}q)^2 + \cdots + (\bar{D}q)^{m-k+1}$$
$$= \frac{D^{m-k+2} - \bar{D}q^{m-k+2}}{D - \bar{D}q} \quad (4)$$

Further, we define the following key notation:

$A(K)$ is the probability that a transmission period is successful, given that there are K ready terminals to start this transmission period.

$NBI(m, K)$ is the average number of packets remaining at the end of a transmission period, given that there are K ready terminals at the beginning of that transmission, and the length of the transmission period is m slots.

$MID(K)$ is the average length of the initial delay period, given that there are K ready terminals at the beginning of the initial delay period.

$NBT(M, K)$ is the average number of ready terminals at the beginning of a transmission period, which is also the average number of ready terminals accumulated at the end of the previous initial delay period, given that MID equals M slots and there are K ready terminals at the very beginning of that initial delay period.

$R(m, K)$ is the probability that there are no ready terminals waiting at the end of a transmission period, there are K ready terminals at the beginning of the transmission period, and the transmission period is m slots long.

From these definitions, we have the following:

Lemma 1:

$$A(K) = K \times \bar{F}p \times (1 - \bar{F}p)^{K-1} \times \left\{1 - \frac{\bar{D}p[1 - (\bar{D}q)^N]}{1 - \bar{D}q}\right\}^{K-1} \prod_{j=1}^{N} e^{-(G/N)HB(N,j)}$$

Lemma 2:

$$NBI(m, K) = K \times \frac{(\bar{D}q)^m}{1 - HB(m, 1)} + \frac{G}{N} \times \sum_{k=1}^{m} \frac{S(m, k)}{1 - HB(m, k)} \quad (5)$$

Proof: To find $NBI(m, K)$, we first find the probability that a packet which is accumulated from the previous initial delay period will stay until the end of the transmission period, given it does not hit the transmission; we then multiply this probability by K. Also, we find this probability for those terminals which become ready during the transmission period, and multiply this probability by G/N, which is the average number of arrivals per slot. Summing up these two values, we obtain $NBI(m, K)$.

$$NBI(m, K) = K \times \text{Prob[stay | not transmit]} + \sum_{k=1}^{m} \frac{G}{N} \text{Prob[stay | not transmit]}$$
$$= K \frac{(\bar{D}q)^m}{1 - HB(m, 1)} + \frac{G}{N} \times \sum_{k=1}^{m} \frac{S(m, k)}{1 - HB(m, k)} \qquad \text{QED}$$

In order to find $MID(K)$, we first define $ID(K)$ to be the length of an initial delay period, given that there are K ready terminals at the beginning of that initial delay period. We will find $\text{Prob}[ID(K) > i]$ for all i from zero to infinity. Obviously, $\text{Prob}[ID(K) > 0] = 1$. For $i > 0$,

$$\text{Prob}[ID(K) > i] = \text{Prob[}K\text{ ready terminals do not transmit for at least } i \text{ slots]} \times \prod_{j=1}^{i} \text{Prob[any packets that arrive in the } j\text{th slot do not transmit for at least } (i - j + 1) \text{ slots])}$$
$$= [HI(i)]K \prod_{j=1}^{N} \left\{\sum_{n=0}^{\infty} \frac{(G/N)^n e^{-(G/N)}}{n!} [HI(i-j+1)]^n\right\}$$

Hence

$$\text{Prob}[ID(K) > i] = \begin{cases} 1 & \text{if } i = 0 \\ HI(i)^K \prod_{j=1}^{i} e^{-(G/N)[1 - HI(i-j+1)]} & \text{if } i > 0 \end{cases} \quad (6)$$

From these results, we have the following:

Lemma 3:

$$MID(K) = \sum_{i=0}^{\infty} \text{Prob}[ID(K) > i] \quad (7)$$

Lemma 4:

$$NBT(M, K) = \text{Max}\left[1, K \frac{F^M - (\bar{F}q)^M}{HI(M-1)(F - \bar{F}q)} + \frac{G}{N} \sum_{j=1}^{M} \frac{F^{M-j+1} - (\bar{F}q)^{M-j+1}}{HI(M-j)(F - \bar{F}q)}\right]$$

Proof: The way to find $NBT(M, K)$ is similar to the way of finding $NBI(m, K)$, as shown in eqn. 5. Moreover, it is clear that unity must be a lower bound for NBT, since each transmission period must be started by at least one ready terminal. QED

Lemma 5:

$$R(m, K) = [1 - (\bar{D}q)^m]^K \prod_{k=1}^{m} e^{-(G/N)S(m,k)} \quad (8)$$

Proof: To find $R(m, K)$ we have to find the probability that all K ready terminals will leave the system (scheduled for retransmission) before the transmission period ends. Further, we also have to find the probability that all terminals which become ready during the transmission will leave the system before the transmission period ends. Hence,

$$\begin{aligned} R(m, K) &= \text{Prob[all } K \text{ ready terminals leave]} \\ &\quad \times \text{Prob[all new coming packets during transmission leave]} \\ &= [1 - (\bar{D}q)^m]^K \text{Prob[all new coming packets during transmission leave]} \\ &= [1 - (\bar{D}q)^m]^K \\ &\quad \times \prod_{k=1}^{m} \left\{ \sum_{n=0}^{\infty} \frac{(G/N)^n e^{-(G/N)}}{n!} [1 - S(m, k)]^n \right\} \\ &= [1 - (\bar{D}q)^m]^K \prod_{k=1}^{m} e^{-(G/N)S(m,k)} \end{aligned}$$

QED

Lemma 6: $L = 1/R(m, K)$.

Proof: Having $R(m, K)$, we know that the probability of a cycle containing exactly k transmission periods is $[1 - R(m, K)]^{k-1} R(m, K)$. We can also find the expected number of transmission periods in a cycle, as $1/R(m, K)$, which is the value of L defined earlier. QED

Having all these lemmas, we are ready to find the system throughput. In order to obtain the average values of the parameters A, B, NBI, MID, NBT and R, we have to uncondition on m, K and M for $A(K)$, $NBI(m, K)$, $MID(K)$, $NBT(M, K)$, and $R(m, K)$. Unfortunately, we cannot find the distributions of these three random variables, m, K and M. Hence, we are forced to make an approximation and to justify the error of this approximation by simulation. The approximation we use is as follows: although we cannot find the distributions of m, K and M, we are still able to find the mean values of m, K and M by the iterative method described in Section 3.2. Denote the mean values of m, K and M to be $\bar{m}$, $\bar{K}$ and $\bar{M}$. We then take $A(\bar{K})$, $NBI(\bar{m}, \bar{K})$, $MID(\bar{K})$, $NBT(\bar{M}, \bar{K})$ and $R(\bar{m}, \bar{K})$ as the values for A, NBI, MID, NBT and R, respectively.

We can now find the throughput using these approximations. In a cycle, there is one idle period, L transmission periods and L initial delay periods. The average length of a transmission period is $(2 - A)N$, which is the parameter 'm' used in eqns. 3, 4, 5 and 8. The probability that a transmission period is successful is A. Since the first transmission period and the first initial delay period in the cycle are different from the other transmission periods and initial delay periods, if we define A_0 to be probability that the first transmission period is successful, and define MID_0 to be the expected length of the first initial delay period, then the throughput S can be obtained as shown in eqn. 1.