RC 20370 (2/6/96)
Computer Science/Mathematics 27 pages

IBM Research Report

The n-ary Stack Algorithm for the Wireless Random Access Channel

Chatschik Bisdikian

IBM Research Division
T. J. Watson Research Center
Yorktown Heights, New York 10598

IBM Research Division
Almaden · T.J. Watson · Tokyo · Zurich

The n-ary Stack Algorithm for the Wireless Random Access Channel

Chatschik C. Bisdikian

IBM Research Division
Thomas J. Watson Research Center
30 Saw Mill River Road, M/S H2-C24
Hawthorne, NY 10532, USA
tel. no.: (914) 784-7439
fax no.: (914) 784-6219
e-mail: bisdik@watson.ibm.com

Abstract: *The emergence of wireless and personal communication networks has brought random access protocols for packet radio networks back to the research forefronts. Most such protocols are based on the ever popular ALOHA protocol. Unfortunately, this protocol is inherently unstable and requires sophisticated schemes to stabilize it. Another class of random access schemes, called limited sensing or stack algorithms, has been proposed that is stable and allows for the dynamic incorporation of new stations into the network. In this paper, we will review the simple to implement n-ary Stack Algorithm, and we will study its performance under various system parameters in the presence of capture, and also in the presence of feedback errors. Finally, we will investigate its maximum system throughput under various traffic generation processes.*

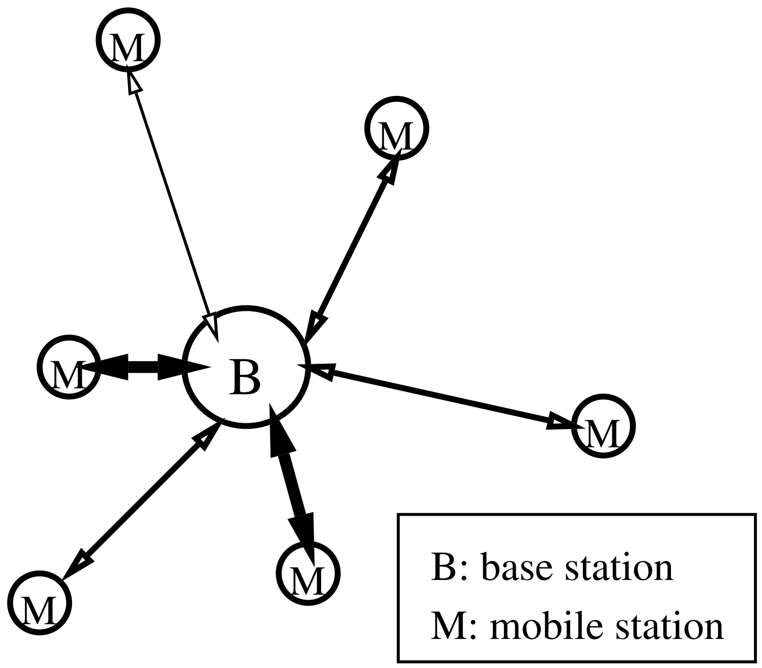

Figure 1: A network topology.

1 Introduction

The recent emergence of wireless, cellular networks has resulted in an increase of proposals for multiple access protocols that regulate the traffic of information in a commonly shared radio channel. For coordinating access on the *up-link* channel, i.e., the communication path from the mobile station to the base-station, see figure 1, most of these proposals use variations of the ALOHA protocol.

ALOHA based protocols are extremely easy to implement and they allow the mobiles to enter and leave the network at will. Yet, infinite population ALOHA networks are well known to be unstable, in the sense that the number of stations that are backlogged (waiting to have a collision resolved) increases to infinity irrespective of the offered load to the network.

Collision resolution algorithms based on tree-search have been devised that are stable, and with throughputs that are higher than what ALOHA is claimed to achieve. But, networks that utilize tree-search algorithms block newcomers, until a current collision resolution terminates, and do not allow for easy incorporation of new mobiles. The tree-research collision resolutions algorithms rely on the history of feedbacks received since at least the last time that the system was empty, which a station cannot know unless it was present throughout the operation of the network; feedback refers to the outcome of a transmission, e.g., collision, success, etc.

Another class of collision resolution random access protocols, called *stack algorithms*, has been introduced that tries to address the previous deficiencies. The stack algorithms are stable with throughputs comparable to tree-search algorithms, they are very simple to implement, and stations can dynamically enter and leave the network, without regard to the feedback history. These algorithms are also called *limited sensing*, because they do not require stations to monitor continuously the feedback channel.

With the advent of wireless communications, limited sensing algorithms have not enjoyed, despite their advantages, the popularity of the ALOHA based protocols. In an effort to change this, in this paper, we will revisit a limited sensing algorithm, called the *n-ary Stack Algorithm* (*nSA*), and analyze its stability under a binary feedback channel and in the presence of capture. In particular, we assume that, say, the base-station in a cellular network informs the stations in its area of coverage, at the end of each transmission opportunity, whether the transmission was successful or not. Moreover, it is assumed that the base-station is capable to decipher correctly one, and possibly more, transmissions even when more than two overlapping transmissions arrive at the base-station; we refer to this capability as *capture*. The base-station notifies the network of the identity of the correctly captured transmission. In case of a packet collision with no capture, the base-station notifies the network that a collision has occurred.

Based on the theory of linear systems with infinite equations and infinite unknowns, in this paper, we will study the throughput of the *nSA* under various generation and capture models, and also under errors in the feedback.

We need to remark here that stack algorithms have also been proposed as part of medium access (MAC) protocols for transmissions over *hybrid fiber/coaxial* (HFC) CaTV networks [3].

The organization of the paper is as follows. In section 2, we will review the network architecture to be considered, and we will introduce a set of transmission phases to support *quality of service* (QoS) oriented services. In sections 3, we will give an overview of the *nSA*, and in sections 4 and 5 and 6, we will highlight its performance under the capture capability. In section 7, we will study the error performance of *nSA*, while in section 8, we will study its performance under various arrival processes. results. Finally, in section 9, we summarize our current work.

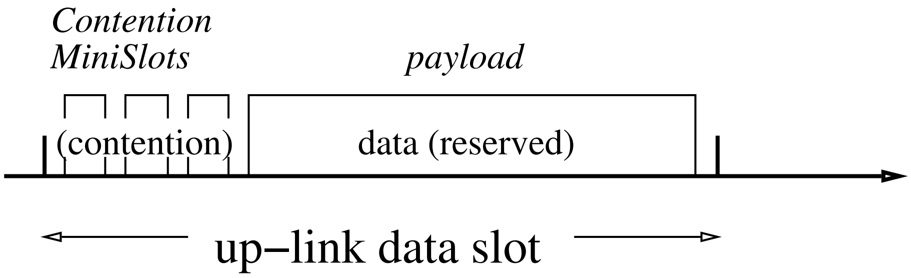

Figure 2: A transmission slot with $K = 3$ CMSs and a data packet (payload) section.

2 Transmission phases for QoS support

In order to achieve high network efficiency and provide acceptable QoS to real-time applications, we believe that the network should operate through a reservations mechanism and be centrally administered. The latter implies that the base-station will have the responsibility to notify which station will transmit what and when. To improve the efficiency further, we believe that waste of network resources due to colliding transmissions should be minimized via the use of *contention minislots* (CMSs). In other words, collision should not occur over the full length of a payload transmission, but rather over much smaller transmission units which we call CMSs.

In particular, we assume that the transmission time is slotted. Each slot, which is of fixed duration, is big enough to contain a fixed size payload section, and K fixed size CMSs. The payload section is used to carry fixed size data packets on contention-free mode, while the CMSs carry data packet reservation information to be transmitted on contention-mode. Stations synchronize their transmissions to the boundaries of the CMSs, whenever they need to access the network with contention-based transmissions, or to the boundaries of the payload sections whenever they access the network with reservation-based transmissions. Figure 2 shows a typical up-link transmission slot with $K = 3$ CMSs and a data payload section.

At the highest level of abstraction for our system, we identify three transmission phases:

1. newly activated stations compete to gain access to the network, using the CMS;

2. if transmissions collide, use a collision resolution algorithm to resolve the collision;

3. following a first successful contention-based transmission, subsequent transmissions are scheduled contention-free based on information provided by the station during earlier transmissions.

Future packet transmissions can be coordinated by the base-station, through a number of mechanisms. The simplest one is utilizing the payload count information P, to be provided as part of the feedback information. The payload count simply gives the number of future payloads that are already reserved. At each slot, the payload count is updated as follows:

- Let S be the number of successful transmissions during a slot, and let $M_1, \ldots, M_S$ be the number of payload requests made during this slot, then

$$P \leftarrow \max\{0, P - 1 + \sum_{i=1}^{S} M_i\}, \tag{1}$$

  where the "-1" above represents any successful payload transmission.

- A station knows when to start transmitting the rest of its traffic by storing the value of P following a successful transmission by the station and counting that many slots.

The above algorithm is not amenable to priority handling of traffic. It is also not flexible in allocating future slots dynamically if QoS constraints of traffic generated after a tagged transmission so demand. Thus, another slot allocation technique may be desirable in such situations, e.g.,

- Prior to each slot the base-station notifies which station should transmit a payload.

This way the base-station may allocate on-the-fly slots to stations that QoS requirements demand them immediately. A slot allocation algorithm, not unlike one found in an ATM switching node, will decide who transmits next. Such an approach is necessary to achieve high degree of efficiency and "QoS fairness," though at the expense of increased complexity at the base-station. A compromise between the two previous approaches is to let the base-station preschedule some of the transmissions as follows:

- The base-station notifies a station that after m slots a station will transmit a number ℓ of packets.

We will not elaborate further on the slot allocation mechanism as its details are beyond the scope of this paper.

Following a first contention-based transmission, the following transmissions from a station can be performed contention-free based on payload reservations. These reservation-based transmissions can persist until all the packets in the stations queues are transmitted. Following this last transmission, the station becomes inactive and future transmissions need to start with a contention-based transmission once again.

Clearly, a most challenging task is coordinating the contention-based transmissions on the CMSs. In the rest of this paper, we will focus on this task and study the *nSA* collision resolution algorithm. Without lack of generality, we will not consider the CMSs as "special" entities anymore. We will study *nSA* on the default slotted ALOHA network model, where a large number of stations synchronize their (data) packet transmissions on the boundaries of fixed size slots. Stations receive feedback regarding the status of a slot, e.g., success, collision, etc, by the end of the slot.

3 The n-ary Stack Algorithm

While the reservation-based transmissions are "well behaved" and under the control of the slot allocation mechanism at the base-station, the first transmission after a station becomes active needs to be contention-based. A number of collision resolution algorithms have been presented in the literature, see [4, 12] and references therein for a wide collection of such algorithms. We have chosen to use the n-ary Stack Algorithm (nSA) because of its limited feedback sensing property and its stable throughput. The former property allows newly activated stations to be dynamically added during an on-going collision resolution without any prior knowledge of the system history; the latter property assures that as long as the generation rate of new traffic is below a *positive* threshold λ^* all packets that participate in a collision resolution are eventually transmitted successfully. Note that the latter desirable property is not shared by the implementations of ALOHA that employ either constant or exponentially backed-off transmission probabilities [1].

According to the rules of nSA, a station with a packet to transmit executes a very simple algorithm that requires the use of a counter C for each packet waiting for a contention-based transmission in a station. For each new such packet, its counter is initiated at $C = 0$.

- When $C = 0$, the station is permitted to transmit in the next slot.
- If a collision occurs, then
  - if $C = 0$, $C \leftarrow Rand\{0, 1, \ldots, n-1\}$, where the operator $Rand\{\cdot\}$ implies the random selection of one of the elements in the braces;
  - if $C \geq 1$, $C \leftarrow C + n - 1$.
- If a no-collision occurs, then
  - if $C = 0$, the station that transmitted (if any) has a successful transmission;
  - if $C \geq 1$, $C \rightarrow C - 1$.

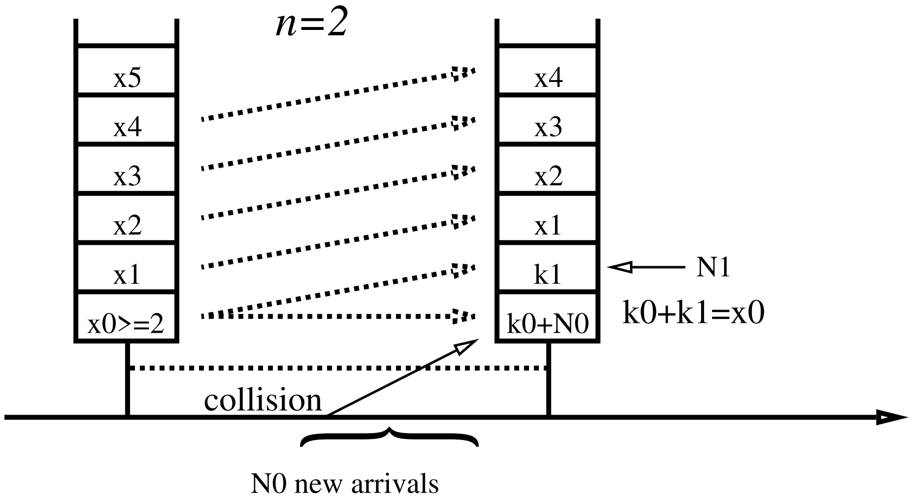

Figure 3: State of the stack before and after a collision.

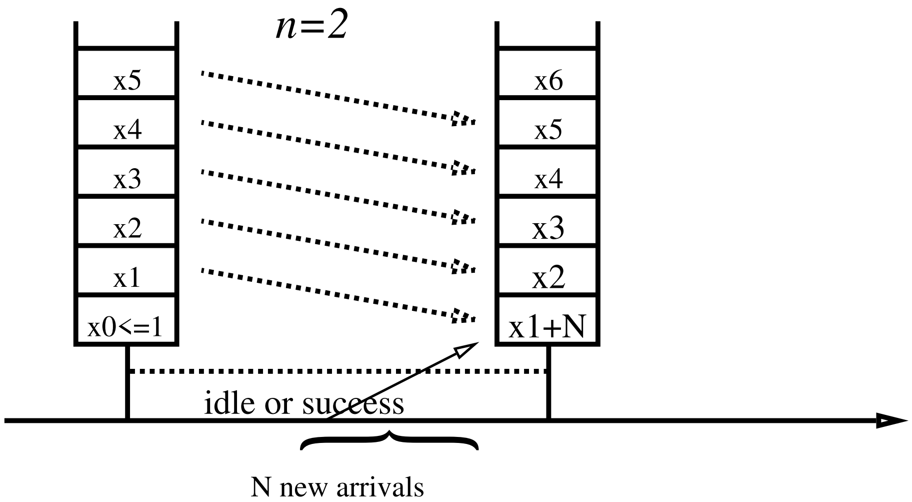

Figure 4: State of the stack before and after a non-collision slot.

The operation of the nSA is best visualized with the help of an imaginary stack with cells numbered as cell $\#0$, $\#1$, etc. Cell $\#i$ contains all the packets in the system whose counters have currently the value i; cell $\#0$ is the transmission cell. Following the rules of the nSA, packets move up and down the imaginary stack until they are successfully transmitted. Figures 3 and 4 depict the status of the imaginary stack before and after a collision or a non-collision slot.

Note that the nSA operates under a binary Collision/No_Collision feedback. A version of the stack algorithm that operates under ternary feedback (Collision/Success/Empty) as well as optimizations of the splitting parameter n have been presented in [9]. Since the performance improvements of these versions are only marginal, they will not be further considered here.

Throughput and delay analyses have been presented in [6, 9, 10]. Here we will present some extensions to these analyses to capture some additional aspects of the protocol that could be present in wireless networks. We will firstly consider variations of nSA when the system has capture capabilities, while later will study the error performance of the algorithm. Finally, we will study the maximum throughput of nSA under various arrival processes.

4 Capture

Due to variations of the transmitted signal power levels, or due to proximity of stations to the base-station, or due to code-division multiplexing techniques, etc, the base-station may be able to *capture*, i.e., decipher successfully a packet transmission even in the presence of conflicting transmissions. Moreover, with multiple receivers, the base-station could even receive several packets simultaneously. In this paper, we will focus mostly on the capture of at most one packet case.

When a packet is captured, the base-station needs not only to acknowledge the successful transmission, but also provide the id of the station that transmitted the captured packet. In the presence of capture, the rules of nSA are slightly modified depending on the feedback information provided by the base-station. We consider three alternatives. For the first two, we assume that the base-station can distinguish a single, and thus successful, packet transmission from a captured one among several transmissions. This information is also transmitted as feedback to the stations at the end of a slot. This feedback model is referred to as *feedback with capture* in [5]. The "lucky" station withdraws this packet from the stack. All other stations that transmitted, by not seeing their own transmission been acknowledged, infer that the successful transmission was not their own packet. So these stations, assume that a collision could have happened, if there were no capture capability, and react as if a collision really happened, and, hence, they choose to place their packet in one of the first n cells of the stack, in other words:

Variation 1:

- operate like in the regular nSA, except
- following capture, packets that were transmitted and did not get acknowledged, select a cell from 0 to $n-1$ of the stack with equal probability;
- following capture, $C = C + n - 1$, for all non-transmitted packets.

Variation 2:

- operate like in the regular nSA, except
- following capture, packets that were transmitted and did not get acknowledged, select a cell from 0 to $m-1$, $1 \leq m < n$, of the stack with equal probability;
- following capture, $C = C + m - 1$, for all non-transmitted packets.

Note that for $m = n$, the two variations coincide. Variation 2 is motivated by the observation, that if a captured packet is removed from its group of transmissions, the remaining packets may not be sufficient to warrant an n-ary splitting. This variation can be further generalized by providing a different splitting parameter n_i if it is revealed the i packets are captured.

For the third variation, we assume that the base-station cannot distinguish a single packet transmission from a captured one among several transmissions. In its feedback, the base-station provides merely the id of the station with the successful transmission. Hence, packets that also transmitted learn implicitly that there was a collision, while packets that did not transmit, consider the last slot as a non-collision slot.

Variation 3:

- operate like in the regular nSA, except
- following capture, packets that were transmitted and did not get acknowledged, select a cell from 0 to $m-1$, $1 \leq m \leq n$, of the stack with equal probability;
- following capture, $C = C - 1$, for all non-transmitted packets.

5 The system model

We consider a slotted time transmission system, with station transmitting to a base-stations (on the *up-link channel*) fixed size packets large enough to fit in one slot. Packet transmissions are synchronized with the slot boundaries. We consider a system with a very large, unspecified number of stations, whose *cumulative* packet generation process is described by the probability mass function

$$g_m = \Pr\{\, m \text{ packets are generated in a slot}\,\}; \tag{2}$$

the packet generation process is assumed to be memoryless from slot to slot. The average arrival rate will be denoted by λ and it is given by

$$\lambda = \sum_{m=0}^{\infty} m g_m. \tag{3}$$

The transmission channel is considered errorless and the only source of errors are overlapping packet transmissions from different stations to the base-station. It is assumed that the base-station has a capture capability, hence, when overlapping transmissions occur the base-station may still be able to capture, i.e., recover successfully, a packet. The capture capability is described by the collection of probabilities, $i \geq 0$, $j \geq 1$,

$$c_{i,j} = \Pr\{\, i \text{ packets are captured} \mid j \text{ overlapping transmissions}\,\}, \quad \sum_{i=0}^{j} c_{i,j} = 1. \tag{4}$$

Note that, although, the case of multiple packet recoveries will not to be studied in detail here, it is not necessarily excluded, and we will provide the appropriate analysis framework whenever possible.

Without lack of generality it is assumed that $c_{0,0} = 0$ and $c_{1,1} = 1$. For a system without capture, $c_{0,j} = 1$ for all $j \geq 2$. Finally, the capture events are assumed to be independent from slot to slot.

Following the capture of a packet, the base-station acknowledges (on the *down-link* channel) at the end of a slot the identity of the station that successfully transmitted at the last slot. In any other event, no acknowledgment is sent. Hence, unless the base-station acknowledges the particular station, a station that has transmitted a packet in a slot knows by the end of that slot that its transmission has resulted in a collision and not recovered by the base-station.

According to variations 1 and 2 of the nSA, the packets in cell $\#i$ of the stack do not get transmitted unless first all packets in cells $< \#i$ do so. Hence, the nSA provides a form of *last-come-first-served* (LCFS) service to the packets. This observation will be used in the next session to simplify the analysis of the protocol by studying the transmission of packets in one cell at a time. Analysis of variation 3 is still a challenging open research topic.

6 Performance analysis

We will study the system based on the theory of regenerative processes. We will identify the stability of the system by discovering infinitely many time instances that the system stochastically restarts itself [10].

Let us assume that at the end of a slot all previously generated packets have been successfully transmitted; we exclude any newly generated packets during this slot that have not yet attempted transmission. Let us place a pointer at cell #1 and let us assume that the pointer behaves exactly like a station whose packet resides at cell #1 of the stack. Let ℓ_j, $j \geq 0$, denote the length in number of slots between the empty slot just defined and the first slot that the pointer "drops" to cell #0, conditioned on the event that at the first slot following the (original) empty slot there are j new packets attempting transmission. The interval, as defined previously, will be called the *session*, and if a session starts with j contenting packets, the session will be call a session of *multiplicity* j, or simply a j-session, $j = 0, 1, 2, \ldots$. It follows from the rules of nSA, see also figures 3 and 4, that[1]

$$\ell_0 = \ell_1 = 1, \quad \text{and} \tag{5}$$

$$\ell_j \stackrel{\text{dst}}{=} 1 + \Big(\sum_{i=0}^{n-1} \ell_{(k_i+N_i)} I\{k_i + N_i\}\Big) I\{\text{no\_capture}\} + \Big(\sum_{i=0}^{\hat{n}-1} \ell_{(k_i+N_i)} I\{k_i + N_i\}\Big) I\{\text{capture}\}, \tag{6}$$

where k_i denotes the number of collided packets in the first slot of the session that select cell $\#i$ of the stack; N_i denotes the new arrivals just prior the beginning of the *subsession* associated with the transmission the k_i packets; and $\hat{n}$ is a random variable that denotes the splitting parameter n following the capture of j packets, $j = 0, 1, 2, \ldots$. The function $I\{\cdot\}$ represents the indicator function of the expression in the braces. Note that due to the symmetries of the system, the random variables $k_i + N_i$, $0 \leq i \leq n-1$ (or $\hat{n} - 1$), are identically distributed, although not independently.

Let $L_j = E(\ell_j)$ be the average value of ℓ_j. First of all, when 0 or 1 packet is transmitted at the beginning of a session, a new session starts a slot later, hence,

$$L_0 = L_1 = 1. \tag{7}$$

If $j \geq 2$ packets transmit in the beginning of a session, all the packets react as if a collision has occurred with the exception of, say, i captured packets that leave the system, the later happens with probability $c_{i,j}$. Using the fact that the summands in the right-hand-side

[1]The notation $\stackrel{\text{dst}}{=}$ stands for "equal in distribution."

(RHS) of (6) are identically distributed, L_j is equal to 1 plus n times $E(\ell_{(k_i+N_i)} I\{k_i+N_i\})$. Let[2]

$$a^n_{j,k} = n(b_n(k;j) * g_k), \tag{8}$$

where the variable $b_n(k;j)$ stands for the binomial probability

$$b_n(k;j) = \left(\begin{array}{c} j \\ k \end{array} \right) (\frac{1}{n})^k (1-\frac{1}{n})^{j-k}, \tag{9}$$

and provides the probability mass function of the packets that select a particular cell following a collision (or a capture), then, exhausting all possibilities yields:

$$\begin{aligned} L_j &= 1 + \sum_{k=0}^{\infty} \Big[\sum_{i=0}^{j} c_{i,j} a^{n_i}_{j-i,k} \Big] L_k \\ &= 1 + \sum_{i=0}^{j} c_{i,j} [a^{n_i}_{j-i,0} + a^{n_i}_{j-i,1}] + \sum_{k=2}^{\infty} \Big[\sum_{i=0}^{j} c_{i,j} a^{n_i}_{j-i,k} \Big] L_k \\ &\stackrel{\text{def}}{=} \gamma_j + \sum_{k=2}^{\infty} \beta_{j,k} L_k, \quad j \geq 2, \end{aligned} \tag{10}$$

where n_i is the value of the splitting parameter n used when i packets are captured.

The stability of the system is strongly tied with finding conditions under which the system of infinite linear equations of infinite unknowns in (10) has a nonnegative, bounded solution [2, 10]. Such systems have been studied in detail in [7]. We will not elaborate further on the details of the stability analysis here. In the Appendix, we highlight a rigorous procedure for the derivation of sufficient conditions for system stability, and also for the derivation of tight linear bounds for the mean session lengths.

The numerical results in the following section are based only on necessary conditions which still provide us with very reliable performance results with much less effort. The necessary condition states that when the infinite dimensionality system in (10) has a nonnegative, bounded solution, then the solution of the sequence of truncated systems, with the coefficients γ_j and $\beta_{j,k}$ set to 0 for $j,k > N$, tends monotonically to the solution of the infinite system as $N \to \infty$ [7]. Here, the monotonicity implies that if x_i^N is the i-th element of the solution of the N-dimensional truncated system, then

$$0 \leq \cdots \leq x_i^N \leq x_i^{N+1} \leq \cdots \leq x_i^*, \tag{11}$$

where x_i^* is the corresponding element of the infinite dimensionality system.

[2]The symbol "$*$" stands for the convolution operator.

$n = 2$	0.36017	$n = 5$	0.38724
$n = 3$	0.40159	$n = 6$	0.37335
$n = 4$	0.39922	$n = 7$	0.35973

Table 1: Throughput of the nSA with no capture.

6.1 Numerical results

Luckily, relatively small systems of dimensionality $N = 30$ are sufficient to obtain performance results for arrival rates even relatively close to the system stability. While we were looking to find the system maximum throughput, we varied the system dimensionality between $N = 50$ and $N = 80$. Below, we will show some representative results of our analysis.

We start first with stability conditions for systems without capture. Table 1 shows the throughput, actually a tight upper bound, of the nSA for various n's. For these numbers, we have assumed Poisson arrivals for the packets, see later for other arrival processes. The nSA results have been obtained, by searching for nonnegative solutions by truncating ($N = 80$) the system in (10). These numbers coincide with the rigorous but also substantially more elaborate methods used elsewhere in the literature [9]. Clearly, the $n = 3$ case provides the best throughput performance, surpassing the (stabilized) ALOHA (maximum) throughput of $e^{-1} \approx 0.36787$ by 9%. Figure 5 shows the average session length, L, as a function of the arrival rate (Poisson process) for various n. The average session length is give by

$$L = \sum_{j=0}^{\infty} g_j L_j = (g_0 + g_1) + \sum_{j=2}^{\infty} g_j L_j. \tag{12}$$

Finiteness of L implies that the system restarts itself stochastically (or regenerates) infinitely many times and, hence, it is stable.

Assuming now that capture is possible, we will first show the increase of system throughput due to capture. In the sequel, we assume a capture of at most one packet, i.e., $c_{i,j} = 0$ for $i \geq 2$. For simplicity, we will write c_j instead of $c_{1,j}$. For our numerical results, we will assume a geometric capture probability

$$c_0 \stackrel{\text{def}}{=} 0, \quad c_1 = 1, \quad c_j = \alpha c_{(j-1)}, \quad j \geq 2. \tag{13}$$

The parameter α, $0 \leq \alpha \leq 1$, represents a measure of *capture effectiveness*; when $\alpha = 0$, we have the regular no-capture network, while when $\alpha = 1$ perfect capture is possible, i.e., a packet is always captured no matter how many transmissions where performed. Table 2 shows the system throughput of the nSA for $\alpha = 0.5$ and various n; the notation $\lambda_u^a(n)$

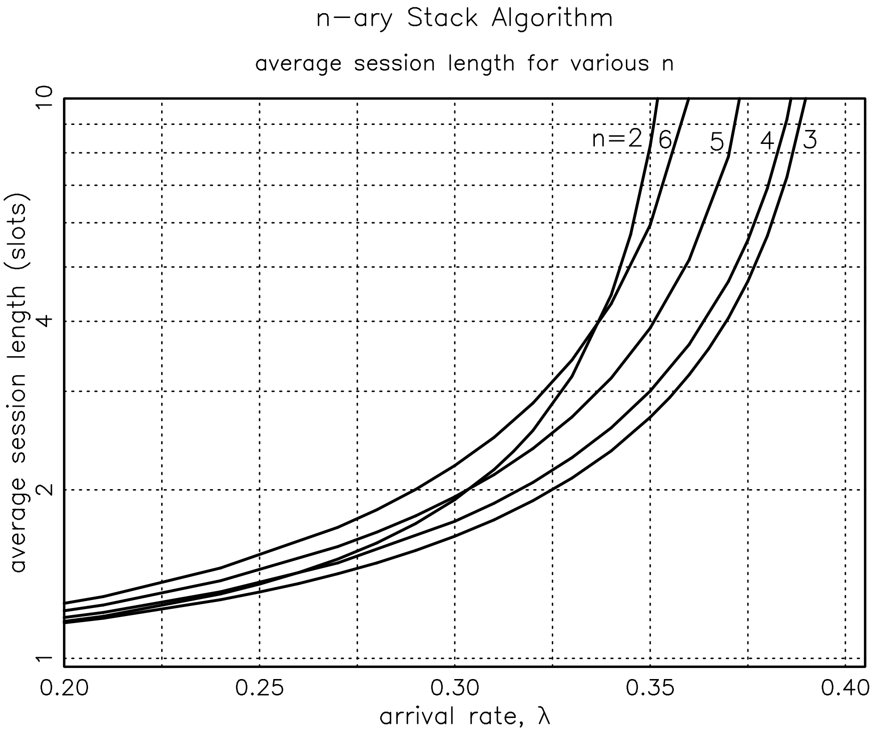

Figure 5: Average session length of the nSA for various n.

for the system throughput is self-explaining, except for the subscript "u" which we use to emphasize that, due to the numerical technique used, these numbers although quite accurate are nevertheless just *upper* bounds to the true system throughput.

We notice that for $\alpha = 0.5$, $n = 3$ is not the best choice any more. Moreover, as n increases the relative improvement due to capture decreases. While for $n = 2$ there is a 46.78% improvement in the system throughput, cf. Table 1, the improvement drops to 15.74% for $n = 7$. This phenomenon is attributed to the fact that most of the collisions are amongst very few packets, two or three. As α increases so does the probability that one of these few collided packets is also captured. Hence, the remaining packets may not benefit significantly by their n-ary splitting among the first n cells of the stack when n is large; this is also demonstrated in Table 4 to be discussed shortly. The $n = 2$ should provide the best selection for increasing n. Figure 6 depicts the average session length for this case.

To investigate further the above observation, Table 3 shows the system throughput for $n = 2$ and 3 for various α. It is clear that the $n = 2$ case represents a more "robust" solution, when it is known that the network has capture capabilities and the capture effectiveness is above a moderate $\alpha \approx 0.3$. These results could also point to adaptive MAC protocols, or other enhancements that would adjust the splitting parameter n to the capture parameters.

n	$\lambda_u^{0.5}(n)$	n	$\lambda_u^{0.5}(n)$
2	0.52867	5	0.46393
3	0.52198	6	0.43845
4	0.49278	7	0.41635

Table 2: Throughput of the nSA for $\alpha = 0.5$ and various n.

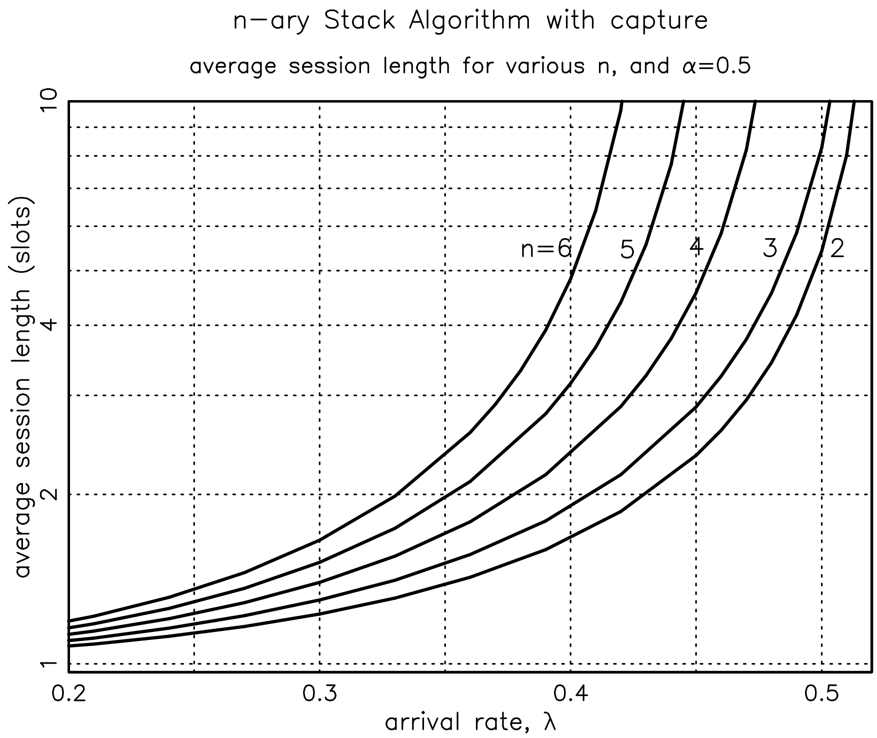

Figure 6: Average session length of the nSA with capture for various n and $\alpha = 0.5$.

α	$\lambda_u^{\alpha}(2)$	$\lambda_u^{\alpha}(3)$	α	$\lambda_u^{\alpha}(2)$	$\lambda_u^{\alpha}(3)$
0.1	0.38810	0.42182	0.7	0.62891	0.58843
0.3	0.45107	0.46744	0.9	0.76088	0.66978
0.5	0.52867	0.52198	1.0	0.83993	0.71651

Table 3: Throughput of the nSA for $n = 2$ and 3 and various α.

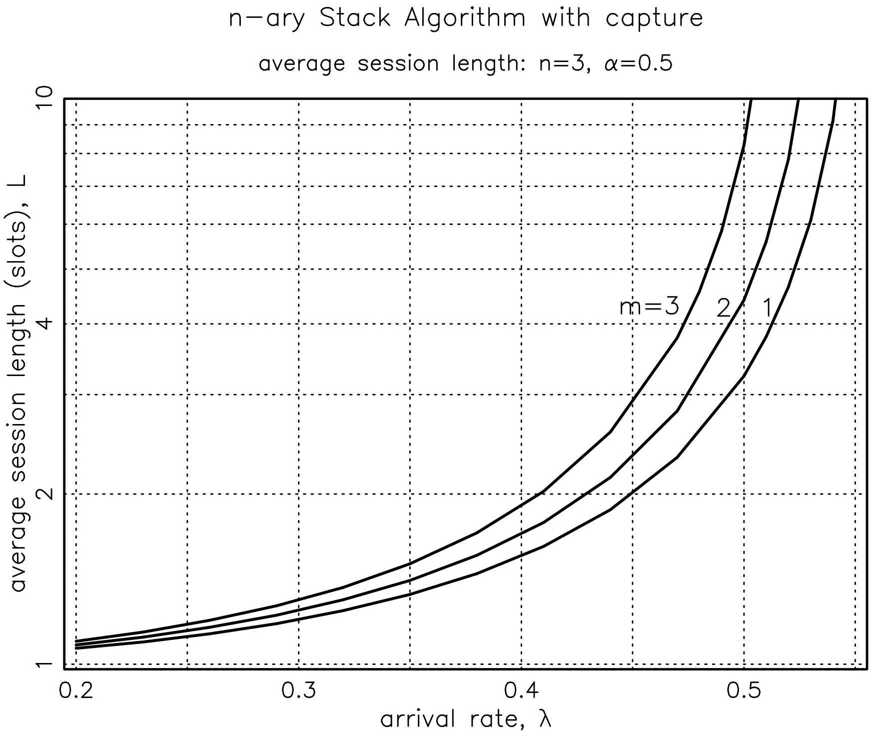

Figure 7: Average session length of the nSA for $n = 3$ and various m.

$n \setminus m$	1	2	3	4	5	6	7
2	0.53697	0.52867					
3	0.55829	0.54300	0.52198				
4	0.54542	0.53043	0.51130	0.49278			
5	0.52615	0.51267	0.49575	0.47929	0.46393		
6	0.50649	0.49460	0.47964	0.46501	0.45125	0.43845	
7	0.48798	0.47751	0.46424	0.45115	0.43876	0.42717	0.41635

Table 4: Throughput of the nSA for $\alpha = 0.5$ and various n and m.

α	$\lambda^{\alpha}_{u,2}(2)$	$\lambda^{\alpha}_{u,3}(2)$	α	$\lambda^{\alpha}_{u,2}(2)$	$\lambda^{\alpha}_{u,3}(2)$
0.1	0.38671	0.38806	0.7	0.52812	0.60141
0.3	0.43688	0.44966	0.9	0.57030	0.69299
0.5	0.48381	0.52046	1.0	0.59073	0.74299

Table 5: Throughput of the nSA for $n = 2$ and truncated capture probabilities.

α	$\lambda^{\alpha}_{u,2}(3)$	$\lambda^{\alpha}_{u,3}(3)$	α	$\lambda^{\alpha}_{u,2}(3)$	$\lambda^{\alpha}_{u,3}(3)$
0.1	0.42104	0.42180	0.7	0.53596	0.57828
0.3	0.45964	0.46684	0.9	0.57387	0.64599
0.5	0.49792	0.51876	1.0	0.59283	0.68312

Table 6: Throughput of the nSA for $n = 3$ and truncated capture probabilities.

The last point has been the motivation for variation 2 in section 4. Table 4 shows results from a recently performed study of variation 2, where following a capture, stations use the splitting parameter m instead of n. It is obvious that selecting $m = 1$ is the best choice for any n, and that once again the selection of $n = 3$ provides for the best throughput performance. We are currently working to capitalize further on this apparent performance superiority of the $(n, m) = (3, 1)$ combination. For the rest of the numerical results, we consider only variation 1.

The capture probabilities in general do not have the nice geometric form, used here for illustration purposes. Depending on the capture parameters of station power strength, distance from the base-station, etc, capture probabilities may have very elaborate forms [8, 13]. Based on our expectation that most collisions are amongst only very few packets, we would like to investigate whether the use of, say, the first couple of the capture probabilities would be sufficient to provide satisfactory throughput approximations. For the next set of results, we use the "truncated" capture probabilities

$$c_0 = 0, \quad c_1 = 1, \quad c_j = \alpha c_{j-1}(1 - u(j - k)), \quad j \geq 2, \tag{14}$$

where $u(n) = 1$ if $n \geq 0$, and 0 otherwise. We study the cases $k = 2$, and $k = 3$, i.e., use only the c_2, or c_2 and c_3 capture probabilities. Table 5 shows the results for $n = 2$ and various α, similarly for Table 6 but for $n = 3$. In these tables, we use the self-explanatory notation $\lambda^{\alpha}_{u,k}(n)$. Comparing these two tables with Table 3, it is apparent that the use of only few of the capture probabilities is sufficient to provide reasonable approximations to the

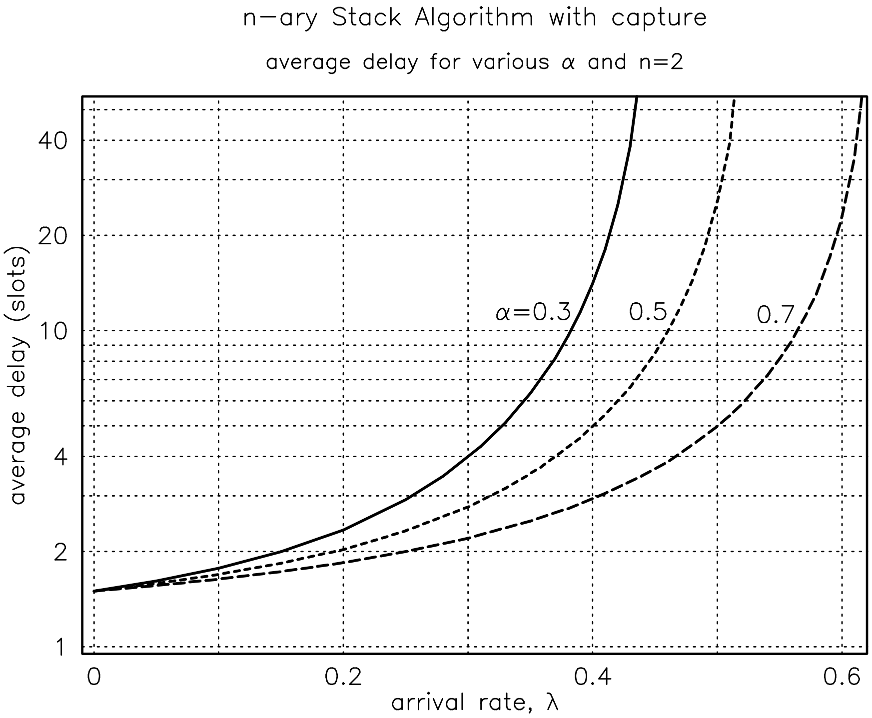

Figure 8: Average delay for the nSA with capture for various α and $n = 2$.

system throughput. For $\alpha = 0.7$, the relative error of $\lambda_{u,3}^{0.7}(2)$ to $\lambda_u^{0.7}(2)$ is -4.37%, and the corresponding number for $n = 3$ is merely -1.72%. Adding three capture probabilities makes even the most stressful case of $\alpha = 1$, which is an upper bound to any capture probability, manageable. In this case, the system capacities are $\lambda_{u,4}^{1}(2) = 0.81308$ and $\lambda_{u,4}^{1}(3) = 0.70998$, with corresponding relative errors -3.20% and -0.91%, respectively. These numbers clearly suggest that good upper and lower bounds to the system performance can be obtained by simply using the first two or three capture probabilities, and then upper (resp. lower) bound the rest with 1 (resp. 0).

Finally, through simulations, we have also studied the delay performance of variation 1 under a Poisson arrival process of packets. The results collected are summarized in figures 8 and 9. Although, no confidence intervals have been developed, the simulation runs were long enough (over 5 million successfully transmitted packets were observed at each simulation run) to guarantee sufficient faith in our results.

Delay is defined as the time elapsed from the moment (during a slot) that a packet is generated up and including its successful transmission time. Hence, assuming Poisson arrivals, the minimum average delay time is 1.5 slots. Figure 8 shows, on a logarithmic scale, the average delay for $n = 2$, while figure 9 shows the average delay for $n = 3$.

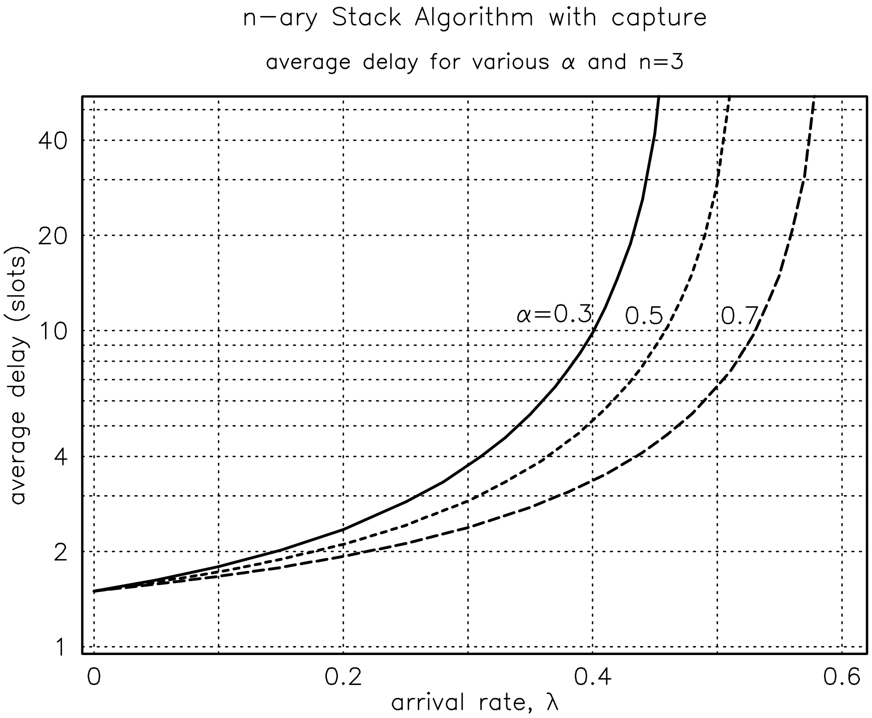

Figure 9: Average delay for the nSA with capture for various α and $n = 3$.

7 Error performance

Our study so far has assumed that no errors occur during the operation of the network. Yet, radio channel interference, noise, hardware deficiencies, etc, could introduce errors in the system. In this section, we will study the performance of the system under a particular error situation, namely the possibility that, on occasions, the base-station cannot distinguish between an empty and a collision slot. In this case, the slot will be declared as a collision slot.

This kind of an error situation was brought to the attention of the author during discussions with physical layer experts. It was disclosed to the author that (physical layer) receiver units on rare occasions just cannot decide on the nature of a non-success transmission. This is one more reason that ALOHA based protocols are so popular, despite their well publishized instability problems. ALOHA based protocols operate only on acknowledgement of successfully transmitted packets. Note that the case a "success→collision" error has been studied in [2], where the stability of nSA is investigated under random transmissions of an interfering station. For simplicity, in the analysis below, we will not consider the capture capability.

n	$\epsilon = 0.00$	0.0001	0.001	0.01	0.05	0.10	0.30
2	0.36017	0.36017	0.36015	0.35995	0.35901	0.35767	0.34930
3	0.40159	0.40159	0.40151	0.40069	0.39673	0.39082	0.34146

Table 7: Throughput of the *nSA* for with probability of error ϵ.

Once again following the approach of the previous section, we can write

$$\begin{aligned} \ell_0 &\stackrel{\text{dst}}{=} 1 + I\{\text{error}\} \sum_{i=0}^{n-1} \ell_{N_i} I\{N_i\}, \quad \ell_1 = 1 \\ \ell_j &\stackrel{\text{dst}}{=} 1 + \sum_{i=0}^{n-1} \ell_{(k_i+N_i)} I\{k_i + N_i\}, \quad j \geq 2. \end{aligned} \tag{15}$$

Note that in this case a session of multiplicity zero does not necessarily end within one slot, but it may last for several slots.

Let $\epsilon = \Pr(I\{\text{error}\} = 1)$, then taking expectations above yields,

$$\begin{aligned} L_0 &= 1 - \epsilon + \epsilon n \sum_{k=0}^{\infty} g_k L_k = 1 - \epsilon(1 - n g_1) + \epsilon n \sum_{k \neq 1} g_k L_k \\ L_1 &= 1 \\ L_j &= 1 + \sum_{k=0}^{\infty} a_{j,k}^n L_k = 1 + a_{j,1}^n + a_{j,0}^n + \sum_{k \neq 1} a_{j,k}^N L_k. \end{aligned} \tag{16}$$

Table 7 summarizes the "empty$\rightarrow$collision" error performance of *nSA*. The robustness of the algorithm, as demonstrated by these results, is remarkable. Even at the high error rate of 10% the throughput achieved is over 97% of the maximum possible.

8 System throughput for non-Poisson arrivals

Once again, for the sake of simplicity, we will consider no capture capabilities. We would like to show that a stability statement of the form

$$\rho < 1, \tag{17}$$

which is so common in queueing theory, can not be directly used for the throughput analysis of random access algorithms.

Note that a random access protocol could be thought of as the server of a queueing system that serves the packets waiting to be transmitted successfully. A statement in the form of (17) implies that the system is stable as long as the *generation rate* of the new packets is below a

M	*binom.*	*geometr.*	M	*binom.*	*geometr.*
2	0.41149	0.39553	8	0.40359	0.39982
4	0.40586	0.39824	16	0.40256	0.40068

Table 8: Throughput of the nSA for $n = 3$ and various arrival processes.

threshold, independently of the actual generation process, except possibly some restrictions on the stationarity properties of the generation process. Yet, this does not seem to apply necessarily to random access protocols, as the following example on nSA demonstrates. All the performance analyses of random access schemes with infinite station population have been done exclusively using the Poisson process as a generation process of new packets. Below we consider two other arrival processes, that have been selected in this demonstration for their simple form.

We consider that the arrival process is a superposition of M Bernoulli processes, i.e., a binomial process, or M geometric processes, each one of mean λ/M for both cases. The z-transform, $G^*(z)$, for the two arrival processes is given by

$$\begin{aligned} G^*(z) &= \Big(1 - \frac{\lambda}{M}(1-z)\Big)^M, \quad \text{(binomial case)} \qquad (18) \\ &= \Big(1 + \frac{\lambda}{M}(1-z)\Big)^{-M}, \quad \text{("geometric" case).} \end{aligned}$$

As M increases and for fixed λ, both cases converge to the Poisson process with rate λ. The coefficient of variation in the binomial case increases to that of the Poisson process, while in the geometric case decreases to that of Poisson. Table 8 shows the throughput of the system for both cases as M increases. Although the differences of the throughput to that of the Poisson process, i.e., $M = \infty$, are not significant (they are in the range of $\pm 2\%$), they are significant enough to demonstrate what was thought so far to be the system maximum throughput may not be so; recently, [11] has demonstrated similar results studying via simulations random access systems with Pareto packet interarrival processes. The system throughput appears to be dependent on the arrival process, and hence a statement of the form of (17) cannot hold true. In our example, the system throughput tends to its limiting value, in a fashion opposite to that of the coefficient of variation.

We believe that the above arrival process dependent system throughput is not a peculiarity of the nSA, but rather it is a property inherent to random access algorithms, especially to those that allow a mixing of newly generated and previously transmitted (unsuccessfully) packets. A collision resolution process of packets is a highly correlated process. An erratic arrival process could produce collision patterns that are very stressing to the system, and a modest increase of the arrival rate could reflect to an abrupt increase of the collision resolution time of packets. What all this points to is that the stability analysis of random access algorithms may need additional scrutiny, in cases that the Poisson arrival process is not an appropriate model for the arrivals.

9 Conclusions

We have studied the performance of the nSA in a system with capture capabilities. The nSA is an inherently stable algorithm that can operate in a limited sensing system. We have developed the mathematical foundation that provides an easy analysis for a plethora of system parameters. We have found that using only few of the capture probabilities provides useful approximations to the system performance. We have also shown that the $(n, m) = (3, 1)$ version of variation 2 gives the best results.

Also based to the flexibility of our mathematical approach, we were able to investigate the throughput of nSA under feedback errors and various arrival processes. For the former, we have demonstrated a high degree of robustness for nSA. For the latter, we have demonstrated that the system throughput is dependent on the arrivals, a property that we believe is shared among the various random access algorithms studied in the literature. These results may motivate further studies in the area in order to capture the effect of (possibly) higher moments of the arrival process on the system throughput.

In this paper, we have not assumed any particular capture model, although a geometric model was used for the numerical results. We believe that the capture probability analysis can be performed independently of the network performance analysis. As long as the capture probabilities have somehow be derived, these can then be used in the network performance analysis.

In future work, we would like to study variations of the transmission procedures following capture as well as possible relaxation of the feedback with capture model, like the variation 3 in section 4. Based on the observation that the $(n, m) = (3, 1)$ selection is the best for variation 2, we now believe that it would be possible to analyze variation 3 as well. Finally, we would also like to develop waiting time analysis tools, by further enhancing our linear systems analysis techniques, as it was done for the non-capture case in [10].

A Bounds on the system performance

Upper bounds on the session length and corresponding lower bounds on the system throughput could be derived the following bounding approach below. We present only the at most one packet captured case.

Consider a sequence $\underline{\mathbf{x}} = \{x_j; j \geq 2\}$ and the transformation $\underline{\mathbf{y}} = \mathcal{T}(\underline{\mathbf{x}})$ defined by

$$\begin{gathered} \underline{\mathbf{y}} = \{y_j; j \geq 2\}, \quad \text{and} \\ y_j = 1 + \beta_{j,0} + \beta_{j,1} + \textstyle\sum_{k=2}^{\infty} \beta_{j,k} x_k, \end{gathered} \tag{19}$$

where $0 \leq \beta_{j,k} \leq n < \infty$, for all j, $k \geq 2$. Moreover, let the following hold true

$$\begin{gathered} \textstyle\sum_{k=0}^{\infty} \beta_{j,k} = n < \infty, \quad \text{and} \\ \textstyle\sum_{k=0}^{\infty} \beta_{j,k} k = j + n\lambda - c_j < \infty, \end{gathered} \tag{20}$$

where $0 \leq \lambda, c_j \leq 1$. Note that for the linear systems considered in this paper, the above conditions are satisfied by definition of the parameters $\beta_{j,k}$, c_j, and λ.

Let us suppose that a linear sequence of the form

$$x_j^{(0)} = Dj - E, \quad j \geq 2, \tag{21}$$

is given, where the coefficients D and E, to be determined later, are independent of j. By applying the $\mathcal{T}$ transformation once, it yields the following

$$x_j^{(1)} = x_j^{(0)} + A + (1+E)(\beta_{j,0} + \beta_{j,1}) - D(\beta_{j,1} + c_j), \quad j \geq 2, \tag{22}$$

where $A = 1 + n\lambda D - (n-1)E$. Let D and E be such that $A = 0$, i.e.,

$$E = \frac{1 + n\lambda D}{n - 1}. \tag{23}$$

Assuming that (23) holds true, applying the transformation once more yields

$$\begin{aligned} x_j^{(2)} &= x_j^{(0)} + (1+E)\Big(\beta_{j,0} + \beta_{j,1} + \sum_{k=2}^{\infty} \beta_{j,k}(\beta_{k,0} + \beta_{k,1})\Big) - D\Big(\beta_{j,1} + c_j + \sum_{k=2}^{\infty} \beta_{j,k}(\beta_{k,1} + c_k)\Big) \\ &= x_j^{(1)} + (1+E) \sum_{k=2}^{\infty} \beta_{j,k}(\beta_{k,0} + \beta_{k,1}) - D \sum_{k=2}^{\infty} \beta_{j,k}(\beta_{k,1} + c_k) \end{aligned} \tag{24}$$

By induction, we obtain in general that

$$x_j^{(m+1)} = x_j^{(m)} + (1+E) Y_j^{(m)} - D Z_j^{(m)}, \quad \text{where} \tag{25}$$

$$Y_j^{(m)} = \sum_{k_1=2}^{\infty} \cdots \sum_{k_m=2}^{\infty} \beta_{j,k_1} \cdots \beta_{k_{m-1},k_m} (\beta_{k_m,0} + \beta_{k_m,1}) = \sum_{k=2}^{\infty} \beta_{j,k} Y_k^{(m-1)}, \quad \text{and} \tag{26}$$

$$Z_j^{(m)} = \sum_{k_1=2}^{\infty} \cdots \sum_{k_m=2}^{\infty} \beta_{j,k_1} \cdots \beta_{k_{m-1},k_m}(\beta_{k_m,1} + c_{k_m}) = \sum_{k=2}^{\infty} \beta_{j,k} Z_k^{(m-1)}; \tag{27}$$

the $m = 0$ case should be interpreted as in (22).

Notice that if, for some $m = M^*$, $0 \le x_j^{(m+1)} \le x_j^{(m)}$ for all $j \ge 2$, then (25) guarantees that $0 \le x_j^{(m+1)} \le x_j^{(m)}$ for all $m \ge M^*$, and hence the sequence $\underline{\mathbf{x}}$ converges pointwise.

The requirement that $x_j^{(m+1)} - x_j^{(m)} \le 0$, for all $j \ge 2$, implies that

$$(1 + E)Y_j^{(m)} - DZ_j^{(m)} \ge 0, \quad \forall j \ge 2. \tag{28}$$

Furthermore, if for an $m = M^*$ (28) holds true, it also holds true for any $m > M^*$, since

$$(1 + E)Y_j^{(m+1)} - DZ_j^{(m+1)} = \sum_{k=2}^{\infty} \beta_{j,k}\Big((1 + E)Y_k^{(m)} - DZ_k^{(m)}\Big), \tag{29}$$

end every element in the RHS is positive for any j and k, confirming our comment in the previous paragraph.

Substituting (23) in (28) yields

$$\begin{aligned} (1 + \tfrac{1+n\lambda D}{n-1})Y_j^{(m)} - DZ_j^{(m)} \ge 0 &\Leftrightarrow (n - 1 + 1 + n\lambda D)Y_j^{(m)} - (n-1)DZ_j^{(m)} \ge 0 \\ &\Leftrightarrow D(n\lambda Y_j^{(m)} - (n-1)Z_j^{(m)}) + nZ_j^{(m)} \ge 0 \end{aligned} \tag{30}$$

Assuming that $n\lambda Y_j^{(m)} - (n-1)Z_j^{(m)} > 0$, D may be selected to satisfy

$$D(m) = \sup_{j \ge 2} \Big\{ \frac{nZ_j^{(m)}}{(n-1)Z_j^{(m)} - n\lambda Y_j^{(m)}} \Big\}, \tag{31}$$

where we have explicitly denoted the dependence of D on m. It follows from (20) that $D(m) < \infty$, for every $m < \infty$.

The stability of the system is governed by the relation

$$n\lambda Y_j^{(m)} - (n-1)Z_j^{(m)} > 0 \Leftrightarrow \lambda > (1 - \frac{1}{n})\frac{Z_j^{(m)}}{Y_j^{(m)}}. \tag{32}$$

Need to caution that the above expression does not imply that the system is stable whenever the arrival rate is larger than some value! Both $Z_j^{(m)}$ and $Y_j^{(m)}$ are functions of λ themselves, actually they are functions of the arrival process distribution.

From previous discussions, it follows that if (32) holds for some λ and m_0, then it holds for the same λ and all $m > m_0$. Hence, if λ_m is the maximum λ for which (32) holds true, the maximum system throughput, λ^*, satisfies

$$\lambda_0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda^*. \tag{33}$$

In general, investigating (32) further requires a numerical approach. Yet in the simple case that $m = 0$, which provides the lowest of the lower bounds on the system throughput could be investigated easily, particularly when no capture is considered. Assuming, say, a Poisson arrival process then

$$Z_j^{(0)} = \alpha_{j,1} = (1 - \frac{1}{n})e^{-\lambda}(\lambda + \frac{j}{n-1}), \quad \text{and} \tag{34}$$

$$Y_j^{(0)} = \alpha_{j,0} + \alpha_{j,1} = (1 - \frac{1}{n})e^{-\lambda}(\lambda + \frac{j}{n-1} + 1), \quad \text{hence,} \tag{35}$$

$$\begin{aligned} &\lambda_0 > (1 - \tfrac{1}{n})\frac{Z_j^{(0)}}{Y_j^{(0)}} \Leftrightarrow \lambda_0 > (1 - \tfrac{1}{n})\frac{\lambda_0(n-1)+i}{\lambda_0(n-1)+i+n-1} \\ &\Leftrightarrow ((1 - \tfrac{1}{n}) - \lambda_0)j - \lambda_0(1 - \tfrac{1}{n})(1 + n\lambda_0) > 0. \end{aligned} \tag{36}$$

Since (36) is increasing in j, it needs to be checked only for its "smallest value," i.e., for $j = 2$, which results in

$$- n(n-1)\lambda^2 - (3n-1)\lambda + 2(n-1) > 0, \quad \text{or} \tag{37}$$

$$\lambda_0 = \frac{1 - 3n + \sqrt{8n^3 - 7n^2 + 2n + 1}}{2n(n-1)}, \tag{38}$$

which coincides with what has been reported in [10].

References

[1] D. J. Aldous, "Ultimate instability of exponential back-off protocol for acknowledgement-based transmission control of random access communication channels," *IEEE Trans. Inform. Theory*, vol. IT-33, no. 2, pp. 219-223, 1987.

[2] C. Bisdikian, L. Merakos, and L. Georgiadis, "Stability analysis of interconnected single-hop random access networks," *IEEE Trans. on Commun.*, vol. 40, no. 3, pp. 556-567, March 1992.

[3] C. Bisdikian, B. McNeil, R. Norman, and R. Zeisz, "MLAP: A MAC Level Access Protocol for the HFC 802.14 Network," *IEEE Communications Magazine*, vol. 34, no. 3, pp. 114-121, March 1996.

[4] C. Bisdikian, "A review of random access algorithms," *Int'l Workshop on Mobile Communications*, pp. 123-127, Thessaloniki, Greece, Sept. 19-20, 1996.

[5] I. Cidon and M. Sidi, "The effect of capture on collision-resolution algorithms," *IEEE Trans. on Commun.*, vol. 33, no. 4, pp. 317-324, Apr. 1985.

[6] G. Fayolle, P. Flajolet, M. Hofri, and P. Jacquet, "Analysis of a stack algorithm for random multiple-access communication," *IEEE Trans. on Inform. Theory*, vol. IT-31, no. 2, March 1985.

[7] L. V. Kantorovich and V. I. Krylov, "Approximate Methods of Higher Analysis," P. Noordhoff, Ltd., 1958.

[8] R. O. LaMaire, A. Krishna, and H. Ahmadi, "Analysis of a wireless MAC protocol with client-server traffic and capture," *IEEE J. Selec. Areas Commun*, vol. 12, no. 8, pp. 1299-1313, Oct. 1994.

[9] P. Mathys and P. Flajolet, "Q-ary collision resolution algorithms in random-access systems with free and blocked channel access," *IEEE Trans. Inform. Theory*, vol. 31, no. 2, pp. 217-243, Mar. 1985.

[10] L. F. Merakos and C. Bisdikian, "Delay analysis of the n-ary stack random access algorithm," *IEEE Trans. Inform. Theory*, vol. 34, part I, no. 5, pp. 931-942, Sept. 1988.

[11] M. Paterakis and Z. Harpantidou, "On the random access of packet broadcast channels with Pareto distributed packet interarrival times," *Int'l Workshop on Mobile Communications*, pp. 234-238, Thessaloniki, Greece, Sept. 19-20, 1996.

[12] R. Rom and M. Sidi, *Multiple Access Protocols: Performance and Analysis*, New York: Springer-Verlag, 1990.

[13] M. Zorzi and R. R. Rao, "Capture and retransmission control in mobile radio," *IEEE J. Selec. Areas Commun*, vol. 12, no. 8, pp. 1289-1298, Oct. 1994.