Notes on the SHUMA Protocol

Scalable Access to Link-16 Time Slots

John Custy, Allen Shum/SSC-SD
{john.custy | allen.shum}@navy.mil
30 March 2006

Summary. This document describes the mechanisms by which SHUMA provides adaptive, scalable access to Link-16 time slots while simplifying network design. Analysis and simulation results illustrate the relative merits of SHUMA versus conventional protocols.

1. Introduction

The Stochastic Unified Multiple Access (SHUMA) channel access protocol provides effective access to Link-16 time slots for fluctuating numbers of users while minimizing collisions. SHUMA is of interest not only for its practical value, but also because it elegantly exploits the situational awareness data found in every Link-16 terminal, and because its functionality can subsume that of conventional, non-adaptive, protocols. The capabilities of SHUMA are of particular interest in light of the steadily increasing numbers of Link-16 users found in theatre, and the increasing flexibility required by those users.

This paper describes SHUMA and illustrates how it achieves its unique performance characteristics, with special emphasis on SHUMA as a generalized form of the contention access protocol. Section 2 reviews the dedicated and contention access protocols, and briefly discusses the geometry of the Link-16 capture effect. The SHUMA protocol is then introduced and some of its characteristics described. Sections 3, 4 and 5 discuss the performance of SHUMA and conventional protocols in terms of time slot usage, average receive intervals, and channel access latency. Section 6 contains some additional commentary and conclusions.

Our discussions will be rather informal, and the reader is advised to consult [Ref. 2] for a more rigorous and comprehensive descriptions of SHUMA. We assume a familiarity with Link-16 at roughly the level of [Ref. 3]. The figures and analysis below were created with *Mathematica* v5.1, and all relevant code not contained in the body of this document is given in the appendix. The reader should thus be able to verify, and potentially extend, all results presented here

2. Link-16 Channel Access Protocols

A *channel access protocol* is a pre-agreed upon arrangement that allows a group of users to share a broadcast communication channel. Mainly to establish terminology, we briefly review the two deployed Link-16 channel access protocols, the dedicated access and contention access protocols.

■ 2.1 Protocols with Fixed Transmission Rates

Because of its conceptual simplicity and the guaranteed performance it provides to users, dedicated access protocol (hereafter denoted "DA") is in many ways ideal. Each user is given exclusive use of a specific set of Link-16 time slots under DA, so multiple transmissions never occur during a single slot, and performance is completely deterministic and thus easy to evaluate.

Unfortunately, DA has a few practical shortcomings. Perhaps most significant is that each terminal must be informed of the unique set of time slots allocated to it before a network can be established. In an operational environment where the number of participants can number in the hundreds, matching each time slot assignment (i.e., terminal load) to the correct terminal is a costly logistic burden that consumes time, reduces flexibility, and, as experience shows, almost inevitably leads to terminal configuration errors. Furthermore, when unanticipated operational requirements, attrition, or other circumstances prevent a user from participating, the slots allocated to that user under DA cannot be recovered.

The contention access protocol (which we'll denote "CA") circumvents this difficulty to some extent. Time slots under CA are partitioned into access intervals, and over the course of each access interval a single transmission is made by each participant. Significantly, each user transmits on a slot within the access interval that is chosen (pseudo-) randomly and independently of any other user.

The average transmission rate for a CA user is fixed at one transmission per access interval, even though the choice of random slot for transmission causes fluctuations about this average. It should be kept in mind that the fluctuations in CA transmission rate are in no way influenced by, or a response to, anything in the environment. Though the particular slot chosen for transmission *changes* from access interval to access interval, this choice is not *driven* in any one way or another by the environment. CA can thus be viewed as a non-periodic form of DA.

The CA protocol reduces/eliminates the logistic burden of DA because multiple terminals operating in the same area can share a common CA load. This is made possible by the random choice of time slots for transmission, which prevents the same user from persistently colliding with each other.

To see why this is important, consider Figure 1. The dark blue dots indicate a hypothetical distribution of ten Link-16 transmitters, and the black lines indicate capture regions associated with each transmitter. (We'll consider the light blue circles and the indices next to each terminal when we discuss ARI in Section 4.) A *capture region*, as the name implies, is that region over which a given transmitter will capture receivers; these regions arise naturally whenever simultaneous transmissions occur, and are not characteristic of any particular protocol.

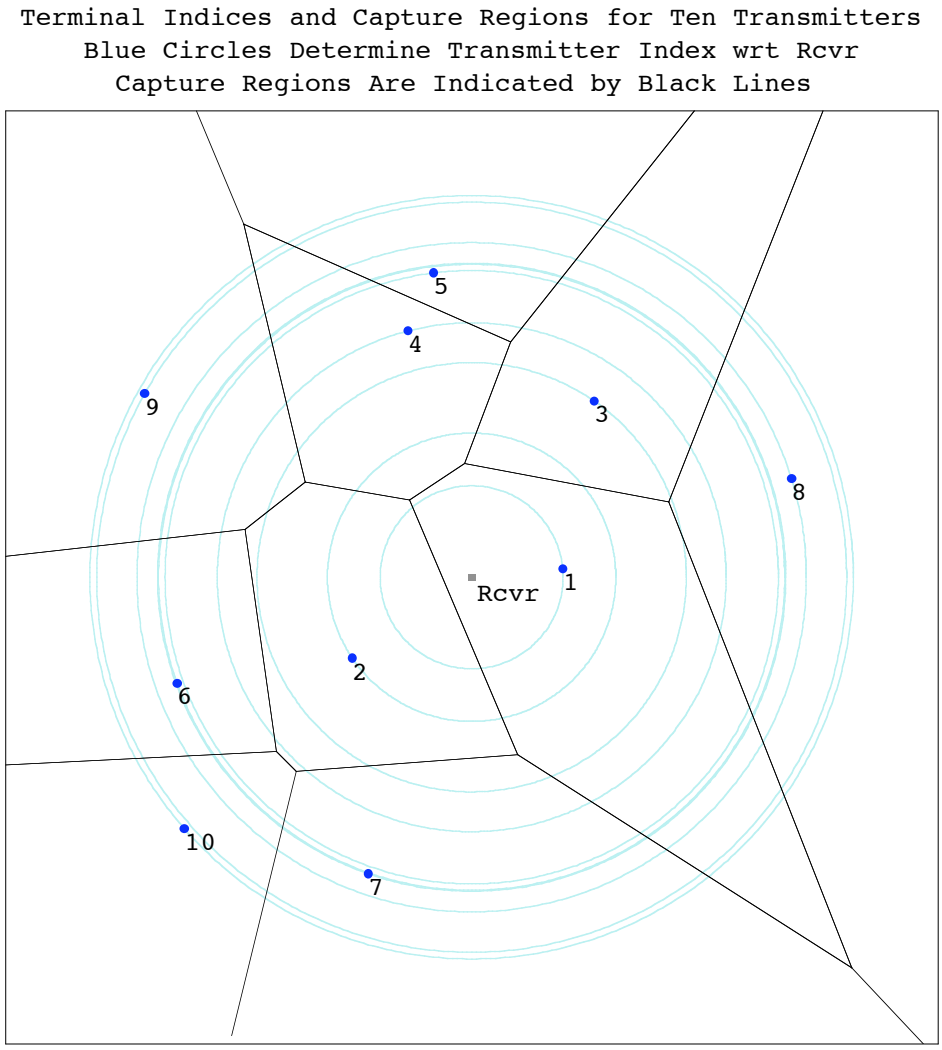


Figure 1. Transmitter indices and capture regions for ten transmitters located at arbitrarily positions around a receiver. The number next to each of the transmitters is an index indicating relative distance from the receiver. The black lines give capture regions when all ten units transmit. Note that in three-dimensional space, terminal indices are determined by spheres, and capture regions are bounded by planes.

Multiple interacting Link-16 units can use a common CA load because, even when platforms are stationary, the pseudorandom choice of slots tends to produce capture regions which change on a slot-by-slot basis. That is, collisions tend to occur among different subsets of users, which results in capture regions which persistently change. This is in contrast to what would happen if the same subset of users persistently transmitted during the same time slot, as might occur if a common DA load were shared by several users. In this case, capture regions would tend to be fixed, and receivers that were stationary with respect to the transmitters would tend to hear the same transmitter persistently.

Though the pseudorandom fluctuations that occur about each users fixed average transmission rate allow capture regions to vary, the fixed average transmission rate nonetheless fixes the average size of capture regions in a crowded environment. Specifically, because CA cannot adjust its parameters in response to operational circumstances, as the number of participants increases beyond the number of slots in an access interval, the number of collisions increases and the size of capture regions decreases. Also, when the number of users happens to be smaller than the number of slots in an access interval, slots will inevitably go unused. In applications where

messages need to be exchanged only among a set of nearest neighbors this performance is acceptable, but the flexibility brought by CA cannot be applied in more general cases in which both close-in and far away units need connectivity.

■ 2.2 SHUMA

CA provides the significant benefit that capture regions change on a slot-by-slot basis regardless of the participants motion. However, all of the parameters of the CA protocol are fixed at network design time and are completely unaffected by operational circumstances. If the parameters of the CA protocol, and in particular the number of slots per access interval, could be dynamically adjusted in response to actual operational conditions, CA could be applied much more extensively: the number of participants in a CA network could be allowed to vary in accord with the dynamic needs of a mission, and both unused time slots and collisions could be held to a minimum.

In general, it's a non-trivial task to adjust the parameters of a channel access protocol as the protocol is being used, but fortunately, we do not have to solve the general problem. One of the primary purposes of Link-16 is to provide users with situational awareness, and the fundamental idea behind SHUMA is to exploit this information, in particular information gained from PPLI exchanges, to regulate terminal transmissions. In a sense, information obtained through PPLIs allows SHUMA to observe, or sense, the environment, and to regulate transmissions accordingly. During tactical operations, situational awareness serves as a significant "force multiplier" that allows collections of individuals to operate as an organic whole, and in a similar manner situational awareness allows networks of SHUMA terminals to behave in ways not possible under DA or CA.

In operation, SHUMA participants each maintain a count of the total participants sharing the time slot pool, a count that is obtained through PPLI exchanges as a normal part of the terminals operations. Before a SHUMA time slot, each terminal generates a random number, and the terminal transmits during the upcoming slot if and only if the random number falls within a range determined, at least in part, by this participant count.

This count of participants is denoted $\mathcal{N}_i(t)$, where the index i indicates a particular terminal, and the argument t reflects the fact that the count can change in operation. In some applications it is useful to place maximum and/or minimum bounds on the count $\mathcal{N}_i(t)$, and these bounds, which are fixed at network design time, are denoted by $\mathcal{N}_{\min}$ and $\mathcal{N}_{\max}$. It is notable that the entire situational awareness picture influences SHUMA through this single parameter.

In addition to the count of participants SHUMA maintains a term, designated $\mathcal{B}_i(t)$, which summarizes the past transmission history of the terminal. This parameter increases throughput for applications in which users do not always have data to send; for example, applications in which users transmit in bursts. The parameter $\mathcal{B}_i(t)$ takes on integer values greater than or equal to zero, and a maximum value, denoted by $\mathcal{K}$, can be set at network design time. In informal terms, $\mathcal{B}_i(t)$ increases when a transmission opportunity is lost because the terminal has no data to send. This increased value of $\mathcal{B}_i(t)$ in turn increases the probability of transmission when data is available, and an actual transmission decreases $\mathcal{B}_i(t)$. A very rough analogy can be drawn between $\mathcal{B}_i(t)$ and a bank account: transmission opportunities that go unused because of lack of data can be stored and later "spent." The reader should consult [Ref. 2, Figure 3] for full information on how $\mathcal{B}_i(t)$ changes in response to data availability and transmission history.

Thus the probability that a SHUMA terminal will transmit is completely specified by the two parameters $\mathcal{N}_i(t)$ and $\mathcal{B}_i(t)$. These two parameters can be thought of informally as the "external" and "internal" parameters: $\mathcal{N}_i(t)$ reflects observations made of the external environment, and $\mathcal{B}_i(t)$ reflects the past transmission history of the terminal. In terms of these two parameters, the SHUMA probability of transmission takes the following form:

$$p_i(t) = 1/\mathcal{N}_i(t) + (1 - 1/\mathcal{N}_i(t))\,(1 - (1 - 1/\mathcal{N}_i(t))^{\mathcal{B}_i(t)}) \tag{1}$$

We can gain some insight into what Equation (1) implies through an approximation and a special case.

This is Equation (1) in two equivalent forms. The "`//`" is postfix notation for the command that follows.

In[1]:=
```
1 / 𝒩 + (1 - 1 / 𝒩) (1 - (1 - 1 / 𝒩)^ℬ) // FullSimplify
```

Out[1]= $1 - \left(\frac{-1 + \mathcal{N}}{\mathcal{N}}\right)^{1+\mathcal{B}}$

Replacing $\mathcal{B}$ with a few small integers suggests that $p_i(t) \approx (\mathcal{B}+1)/\mathcal{N}$ when $\mathcal{N}$ is large. This approximation can be justified more formally using the binomial theorem. The "`%`" represents the immediately preceding result (labeled *Out[1]* above), and the term "`/.`" can be read as "substitute in" or "replace."

In[2]:=
```
% /. ℬ → {0, 1, 2, 3} // Expand // ColumnForm
```

Out[2]=

$$\frac{1}{\mathcal{N}}$$
$$-\frac{1}{\mathcal{N}^2} + \frac{2}{\mathcal{N}}$$
$$\frac{1}{\mathcal{N}^3} - \frac{3}{\mathcal{N}^2} + \frac{3}{\mathcal{N}}$$
$$-\frac{1}{\mathcal{N}^4} + \frac{4}{\mathcal{N}^3} - \frac{6}{\mathcal{N}^2} + \frac{4}{\mathcal{N}}$$

Thus the probability of transmission for a SHUMA terminal can be expressed

$$p_i(t) \approx (\mathcal{B}+1)/\mathcal{N} \text{ for large } \mathcal{N}. \tag{2}$$

Furthermore, in many applications it makes sense to fix $\mathcal{B}_i(t) \equiv 0$, and in this important special case, Equation (1) directly reduces to $p_i(t) = 1/\mathcal{N}_i(t)$.

With this background in mind, we can describe some of the characteristics that Equation (1) implies for SHUMA networks.

2.2.1 Choosing a Value for the Parameter $\mathcal{K}$

As described above, the parameter $\mathcal{B}$ is a non-negative integer that reflects a units transmission history, and the parameter $\mathcal{K}$ is a maximum value for $\mathcal{B}$ set by the network designer. We can get some insight into how these parameters were motivated by considering the case in which a set of participants have, on the average, data available for transmission only some fraction p_{data} of the time.

If a terminal has data to transmit with probability p_{data}, and the SHUMA protocol allows a transmission to occur with probability p_i, then the probability of unit i making a transmission is $p_{\text{data}}\, p_i$. The probability that exactly one out of $\mathcal{N}$ units will make a transmission is given by the following.

In[3]:=
```
𝒩 (p_data p_i) (1 - p_data p_i)^(𝒩-1)
```

Out[3]= $\mathcal{N}\, p_{\text{data}}\, p_i\, (1 - p_{\text{data}}\, p_i)^{-1+\mathcal{N}}$

We'd like to maximize this probability through proper choice of p_i, so we differentiate with respect to p_i and set the result to zero.

In[4]:=
```
D[%, pi] == 0 // FullSimplify
```

Out[4]= $\mathcal{N}\, p_{\text{data}}\, (1 - p_{\text{data}}\, p_i)^{-1+\mathcal{N}}\, (-1 + \mathcal{N}\, p_{\text{data}}\, p_i) == 0$

The command `Reduce` is used here to isolate the term p_i without making any assumptions about other parameters. To minimize spurious solutions, we constrain $\mathcal{N}$ be greater than one, and we constrain the product $p_{\text{data}}\, p_i$ to be a non-trivial probability. The term "`&&`" represents logical AND.

In[5]:=
```
Reduce[% && N > 1 && 0 < pdata pi < 1, pi]
```

Out[5]= $p_{\text{data}} \in \text{Reals}\ \&\&\ p_{\text{data}} \neq 0\ \&\&\ \mathcal{N} > 1\ \&\&\ p_i == \frac{1}{\mathcal{N}\, p_{\text{data}}}$

This result implies that when units always have data to send, the probability of transmission can simply be set to $1/\mathcal{N}$, but when users do not always have data to send, throughput can be increased by adjusting the probability of transmission upward. We can infer two ways for fixing the parameter $\mathcal{K}$.

1. First, if the actual traffic is well modeled by a Bernoulli process, $\mathcal{K}$ can be selected to maximize $p_i(t)$. With the results $p_i = 1/(\mathcal{N}\, p_{\text{data}})$ and $p_i \approx (\mathcal{B}+1)/\mathcal{N}$ (as per Equation (2)) we can infer $\mathcal{B} \approx (1/p_{\text{data}}) - 1$. Thus a single transmission per time slot can be maintained and throughput maximized with $\mathcal{K} \equiv (1/p_{\text{data}}) - 1$.

2. Alternately, if the traffic is not Bernoulli-like, but rather consists of bursts of approximately b messages, it is reasonable to set $K \equiv b$.

2.2.2 Scalability

The adaptive mechanism implied by Equation (1) is very simple, yet it leads to behavior that is very distinct from that observed under static protocols like DA and CA. In particular, SHUMA is *scalable* in the sense that it maintains effective use of the Link-16 channel, more specifically it maintains one transmission per time slot, regardless of the number of participants. Scalability comes about under SHUMA because each user adjusts its transmission rate to maintain on the average one transmission in each time slot; when the number of users increases, each user "backs off" in its transmission rate to allow all users to hear all transmissions. When the number of users decreases, transmission rates can increase correspondingly. Each user "thinks globally and acts locally."

If one transmission per time slot is maintained regardless of the number of participants $\mathcal{N}$, and if transmissions are independent among users, the interval between transmissions from any given user is necessarily proportional to the number of users $\mathcal{N}$. The situation is a simple consequence of the finite number of Link-16 time slots, and cannot be interpreted as a deficiency or weakness introduced by SHUMA. The situation is like waiting in a queue: if the queue is long (many participants sharing a time slot pool), the wait for service will also be long (the interval between transmissions will be large).

Of course, instead of requiring users to "take turns" as per SHUMA, multiple transmissions in each time slot can be allowed as per CA. This is analogous to breaking a long queue up into parts, and providing each "sub-queue" with a separate server; unfortunately for Link-16 users, the result is a decrease in service quality which may not be acceptable for all applications. In particular, multiple transmissions per time slot limits the distribution of messages.

It's interesting to consider ways in which similar adaptive behavior might be obtained in a general purpose TDMA broadcast system, where by "general purpose" we mean a system that does not provide situational awareness to

users. If the information required to regulate transmissions is not a part of the normal channel traffic, the use of some sort of overhead, or "side" information, suggests itself. The channel would then support two distinct types of information: the overhead, and the "real" application information. If overhead increases and decreases with the number of users, as seems inevitable, the channel capacity available for application data could become small or could be eliminated entirely as the number of users sharing the channel increases. SHUMA is able to circumvent this problem because of the dual role played by PPLI messages: these messages serve as both data *and* overhead, and applications for the channel do not have to be compromised simply to maintain the channel.

SHUMA's scalability is thus closely associated with the fact that it requires no explicit exchange of overhead. The dual role played by PPLIs means that the application data *is* the overhead, and so the entire channel can be used for application data without compromising the benefits of scalability.

2.2.3 Spatial Reuse

When groups of users are beyond line of sight of each other, the adaptive nature of SHUMA makes all allocated slots available for each group because members of each subgroup can count only those units within line of sight, and transmissions cannot reach or block any other users. Spatial reuse cannot occur under DA, and occurs under CA only in the weak sense that fewer collisions will occur among smaller groups of users.

To roughly quantify spatial reuse under SHUMA, suppose that there is a total of $\mathcal{N}$ participants deployed, and suppose that because of line of sight constraints an arbitrary SHUMA participant i receives PPLIs from only $\alpha_i \mathcal{N}$ units, where $0 \le \alpha_i \le 1$. Further, suppose that only some fraction $0 \le \beta \le 1$ of participants have data to transmit during any particular time slot. If we consider only that subset of slots in which the participant furthest from unit i makes a transmission, that transmission will be received by unit i with probability

$$\Pr(\mathcal{N}, \alpha_i, \beta) = (1 - 1/\mathcal{N})^{\alpha_i \beta \mathcal{N} - 1}.$$

As $\mathcal{N}$ increases, this probability approaches the following limit from below. That is, if α quantifies line of sight constraints and β quantifies traffic load, then the worst case probability of a "successful" transmission behaves according to the following.

```
In[6]:= Limit[(1 - 1 / N)^(α β N-1), N → ∞]
```

Out[6]= $e^{-\alpha \beta}$

Thus, the probability of a successful transmission within a large group of users ($\alpha \approx 1$) that all have something to say ($\beta \approx 1$) is bounded below by a value of about 37%. If this group is broken up into several smaller groups (each with, say, $\alpha \approx 0$) by line of sight constraints, each subgroup gets full use of channel time slots, increasing the probability of successful transmissions and increasing update rates within each subgroup. Spatial reuse influences the probability of reception in roughly the same way as the traffic load of the users.

2.2.4 Fair Access to Time Slots. A Hierarchy of Protocols

There are no privileged users under SHUMA: each user gets the same average number of accesses to the channel per unit time, and no explicit provisions exist for enhanced or degraded service. At first glance, this might appear to be a painful handicap, but in fact users of different privilege can be easily accommodated through multiple separate or overlapping time slot pools.

This idea can be used to gain some insight into how DA, CA and SHUMA are related to each other. The adaptive nature of SHUMA can be constrained or eliminated through the appropriate choice of values for $\mathcal{N}_{\min}$ and $\mathcal{N}_{\max}$, and SHUMA can be made to simulate the fixed transmission rate of a CA network by setting $\mathcal{N}_{\min} \equiv \mathcal{N}_{\max}$ at the appropriate values.

To illustrate, suppose that s time slots per 12 second frame are allocated under CA with α access intervals per frame. Assuming the same number of slots, we can obtain the same transmission rate under SHUMA by setting the probability of transmission to $p = \alpha / s$, or by setting the parameters $\mathcal{N}_{\min}$ and $\mathcal{N}_{\max}$ to $1/p = s/\alpha$. We'll see below that with this choice of parameters the time slot usage and ARI performance of SHUMA is the same as that of the corresponding CA network.

Just as the parameters of a SHUMA network can be adjusted to mimic any particular CA network, the parameters of a CA network can be adjusted to mimic any particular DA network. At least conceptually, we can create access intervals that contain only one slot, and we can allocate as many or as few of these networks to each user as required. The single time slot per access interval eliminates the influence of the pseudorandom choice of time slot under normal CA, though a Link-16 terminal may not accept the parameters needed to achieve this.

We thus have a hierarchy of Link-16 protocols of increasing generality. DA is a special case of CA in which users relinquish the pseudorandom choice of time slots and instead make periodic transmissions, and CA is a special case of SHUMA in which the length of an access interval is fixed at network design time.

2.2.5 Errors in Each Terminals Estimate of $\mathcal{N}$. Stability

The parameter $\mathcal{N}_i(t)$ within each terminal is only an estimate of the true value of $\mathcal{N}(t)$, and because of the less than ideal environments in which Link-16 networks must operate (jamming, dynamic line of sight conditions, frequent entry and exit of units), it is important that a protocol like SHUMA be robust to errors.

To see that SHUMA is not sensitive to errors in its observations of the environment, we'll consider the stressful case in which users always have data to send, which allows us to set $\mathcal{B} = 0$. (Situations in which users do not always have data to send are less stressful and should be considered on a case-by-case basis.) Suppose that a terminal estimates $\mathcal{N}$ at a value that is in error by ε, so that the correct value is $\mathcal{N}$ and the estimated value is $(1 + \varepsilon)\,\mathcal{N}$. We can use this notation with Equation (1) to find out what happens when all units overestimate or underestimate the true count $\mathcal{N}$.

Assuming $\mathcal{B} = 0$, if the number of terminals within an area is $\mathcal{N}$, but a terminal mis-estimates this count as $(1 + \varepsilon)\,\mathcal{N}$, the optimal and the actual probabilities of transmission will differ as follows. The semicolon at the end of the line prevents it from being printed.

In[7]:= $$\left\{p_{\text{opt}} \to \frac{1}{\mathcal{N}},\ p_{\text{act}} \to \frac{1}{(1+\varepsilon)\,\mathcal{N}}\right\};$$

The SHUMA probability of transmission was chosen to maximize the probability that there will be a single transmission in each time slot. When a sub-optimal p is used, the probability of a single transmission will necessarily decrease. The following gives a normalized measure of this decrease.

In[8]:= $$\frac{\mathcal{N}\,p_{\text{opt}}\,(1-p_{\text{opt}})^{\mathcal{N}-1} - \mathcal{N}\,p_{\text{act}}\,(1-p_{\text{act}})^{\mathcal{N}-1}}{\mathcal{N}\,p_{\text{opt}}\,(1-p_{\text{opt}})^{\mathcal{N}-1}}\ \text{/. \% // FullSimplify}$$

Out[8]= $$1 - \frac{(-1+\mathcal{N})^{1-\mathcal{N}}\,\mathcal{N}^{\mathcal{N}}\,\left(1 - \frac{1}{\mathcal{N}+\mathcal{N}\,\varepsilon}\right)^{\mathcal{N}}}{-1+\mathcal{N}+\mathcal{N}\,\varepsilon}$$

This expression becomes simpler when $\mathcal{N}$ is large. Though the computations are not shown here, the first derivatives with respect to ε of `Out[8]` and `Out[9]` are both zero when ε is zero.

In[9]:=
```
Limit[%, 𝒩 → ∞] // FullSimplify
```

Out[9]= $1 - \frac{e^{\frac{\varepsilon}{1+\varepsilon}}}{1+\varepsilon}$

The limit for large $\mathcal{N}$ is shown here in black, and results for finite $\mathcal{N}$ are in color.

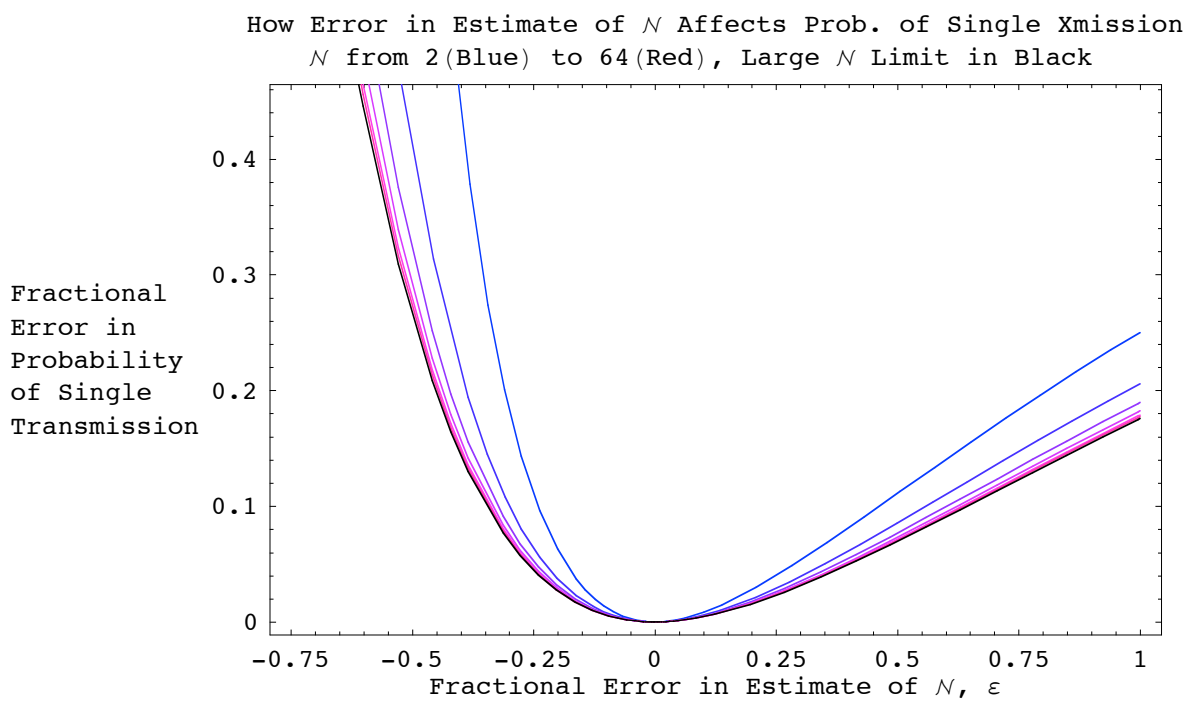


Figure 2. Influence of errors in $\mathcal{N}$ on the probability of single transmission. As an example of how this figure should be read, the black curve shows that when all terminals in a large network under-estimate $\mathcal{N}$ by 25%, the probability of a single transmission changes by about 10% from its nominal value. The curves are for values of $\mathcal{N}$ equal to 2 (blue), 4, 8, 16, 32, and 64 (red), with large $\mathcal{N}$ in black.

Another point of interest is that when SHUMA is used to regulate transmissions of PPLIs, a feedback control loop is formed: SHUMA uses information derived from PPLIs to regulate transmission of PPLIs. However, there are no significant issues regarding stability (for example, terminals that never transmit or that persistently transmit) because the observable used by SHUMA, namely the count $\mathcal{N}$, influences the transmission behavior *algebraically* through Equation (1), and not through a differential or difference equation.

3 Time Slot Usage

Under both CA and SHUMA, it is inevitable that during some time slots no transmissions will occur, and during other time slots two or more transmissions will occur. Even under DA, circumstances unforeseen at network design time (such as changes in force composition due to attrition, terminal load errors, etc.) will cause fewer than one transmission per time slot. The term *time slot usage* will be used to refer to the fraction of time slots in which zero, one, two, …, transmissions occur, and in this section we'll compare CA and SHUMA in terms of this performance measure.

■ 3.1 Contention Access Time Slot Usage

An analytic expression for slot usage under CA can be established by considering a time slot at an arbitrary location in an access interval. If we assume no knowledge of what each participant did in any previous time slots, and if we assume no knowledge of what each participant will do in following time slots, and if we assume that each access interval contains s time slots, then we can say that the probability of transmission by any particular participant is simply $1/s$, and as a result the probability of exactly $\Bbbk$ out of $\mathcal{N}$ participants transmitting is given by the following expression.

$$\Pr_{\mathrm{CA}}(\Bbbk, \mathcal{N}, s) = \binom{\mathcal{N}}{\Bbbk} (1/s)^{\Bbbk} (1 - 1/s)^{\mathcal{N}-\Bbbk} \tag{3}$$

This equation makes sense: because the transmission behavior of any one unit can have no effect on the transmission behavior of any other unit, we can say that the probability of a *specific* set of $\Bbbk$ users transmitting while the remaining $\mathcal{N} - \Bbbk$ are silent is $p^{\Bbbk}(1-p)^{\mathcal{N}-\Bbbk} = (1/s)^{\Bbbk} (1 - 1/s)^{\mathcal{N}-\Bbbk}$. However, we don't care which $\Bbbk$ users are transmitting, so we multiply by a term that represents the number of distinct subsets of size $\Bbbk$ that can be taken from a group of size $\mathcal{N}$.

For fixed values of s and $\mathcal{N}$, we can plot Equation (3) as a function of $\Bbbk$, the number of transmissions in a time slot. Figure 2 shows several such plots, one for each value of $\mathcal{N}$ from 2 to 64 in steps of two, and with the number of slots in each access interval fixed at 16. The special case in which the number of users $\mathcal{N}$ equals the number of slots in an access interval s is plotted in black. Of course, the parameter $\Bbbk$ can take on only integer values, but it is plotted as a continuous parameter in Figure 3 to make behavior as clear as possible as $\mathcal{N}$ changes.

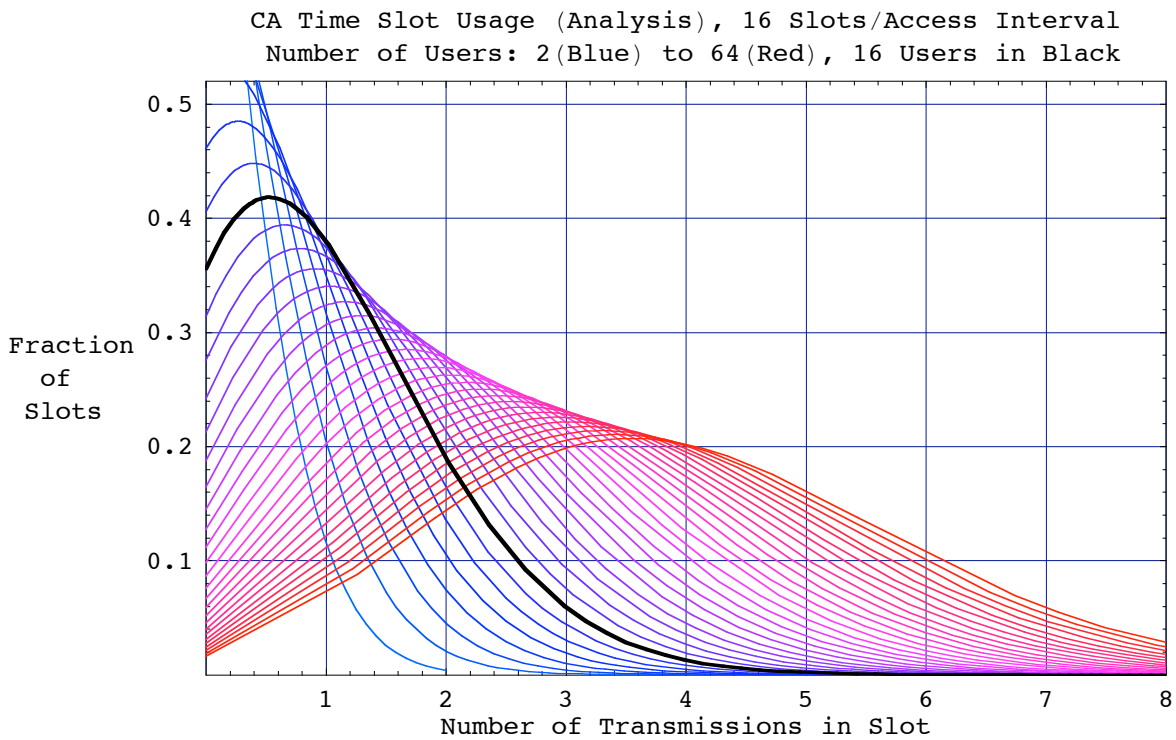


Figure 3. Time slot usage for contention access with 16 slots per access interval and with number of users ranging from 2 to 64 in steps of two. These curves correspond to Equation (3). The case of 16 users, in which each access interval contains one time slot for each user, is shown by the black curve.

As intuition suggests, and as Figure 3 illustrates, the number of transmissions that occur during any CA time slot is closely linked to (a) the number of time slots in an access interval, which is fixed at network design time; and (b) the number of users, which can change during the course of operations. If the number of users is less than the number of slots per access interval, some slots will inevitably go unused, but two or more transmissions during a time slot will be relatively rare. On the other hand, if there are more users than slots per access interval, multiple transmissions during a time slot will be inevitable, though valuable slots will rarely go unused.

Of particular interest is that the heavy black curve in Figure 3 maximizes the fraction of time slots in which a single transmission occurs. That is, this curve provides the point on the "envelope," or the loci of maximum values, where one transmission per slot occurs. As intuition might suggest, the figure indicates that when the number of users equals the number time slots in an access interval, the fraction of time slots that contain a single transmission is maximized.

■ 3.2 Time Slot Usage Under SHUMA

Using exactly the same reasoning that gave us Equation (3), we find that the following expression gives the probability that exactly $\Bbbk$ out of $\mathcal{N}$ SHUMA participants will transmit in a SHUMA time slot.

$$\Pr_{\text{SHUMA}}(\Bbbk, \mathcal{N}) = \binom{\mathcal{N}}{\Bbbk} p^{\Bbbk} (1-p)^{\mathcal{N}-\Bbbk}, \text{ with } p = 1 - (1 - 1/\mathcal{N})^{1+\mathcal{B}} \tag{4}$$

Though this expression is complicated when the substitution for p is explicitly carried out, it simplifies considerably when $\mathcal{N}$ is large.

Equation (4) represents the probability that $\mathbb{k}$ participants will transmit during a particular time slot. When we replace every occurrence of p with the general SHUMA probability of transmission we get the following...

In[10]:= $\frac{\mathcal{N}!}{(\mathcal{N}-\mathbb{k})!\,\mathbb{k}!}\, p^{\mathbb{k}}\,(1-p)^{\mathcal{N}-\mathbb{k}}$ `/. p → 1 - (1 - 1 / N)^(1+B) // FullSimplify`

Out[10]= $\frac{\left(1-\left(\frac{-1+\mathcal{N}}{\mathcal{N}}\right)^{1+\mathcal{B}}\right)^{\mathbb{k}}\left(\left(\frac{-1+\mathcal{N}}{\mathcal{N}}\right)^{1+\mathcal{B}}\right)^{\mathcal{N}-\mathbb{k}}\mathcal{N}!}{(\mathcal{N}-\mathbb{k})!\,\mathbb{k}!}$

...which becomes simpler when $\mathcal{N}$ is large. As before, the term "`/.`" above means "replace all" and the symbol "`%`" represents the immediately preceding expression.

In[11]:= `Limit[%, N → ∞]`

Out[11]= $\frac{e^{-1-\mathcal{B}}\,(1+\mathcal{B})^{\mathbb{k}}}{\mathbb{k}!}$

Remarkably, when a large number of participants share time slots under SHUMA, time slot usage follows a Poisson distribution and is independent of $\mathcal{N}$. We can immediately infer that when *Out[11]* holds, that is, when $\mathcal{N}$ is large, the average number of transmissions per time slot will be $\lambda = \mathcal{B} + 1$, and the standard deviation of the number of transmissions per time slot will be $\lambda = \mathcal{B} + 1$.

Because the value of $\mathbb{k}!$ grows faster than $(1+\mathcal{B})^{\mathbb{k}}$ as $\mathbb{k}$ gets large, it is unlikely in general that a large number of transmissions will occur during a single time slot, even when the number of participants is large. However, this is not because the terminals stop transmitting: the value at $\mathbb{k} = 1$ does not go to zero, but rather decreases to the value $(\mathcal{B}+1)\,e^{-(\mathcal{B}+1)}$.

Figure 4 indicates time slot usage under SHUMA for various values of the parameter $\mathcal{N}$. Significantly, as $\mathcal{N}$ becomes large, the fraction of slots with one transmission *decreases* to its final value, and the fraction of slots with zero transmissions *increases* towards its final value. That is, performance for finite $\mathcal{N}$ tends is always better than when $\mathcal{N}$ is large, and the $1/e \approx 36.7\,\%$ probability of single transmission is a *worst case* bound that will not be experienced in networks of finite size. When a finite number of users share a pool of time slots under SHUMA, a single transmission occurs in *more* than 36.7% of the slots.

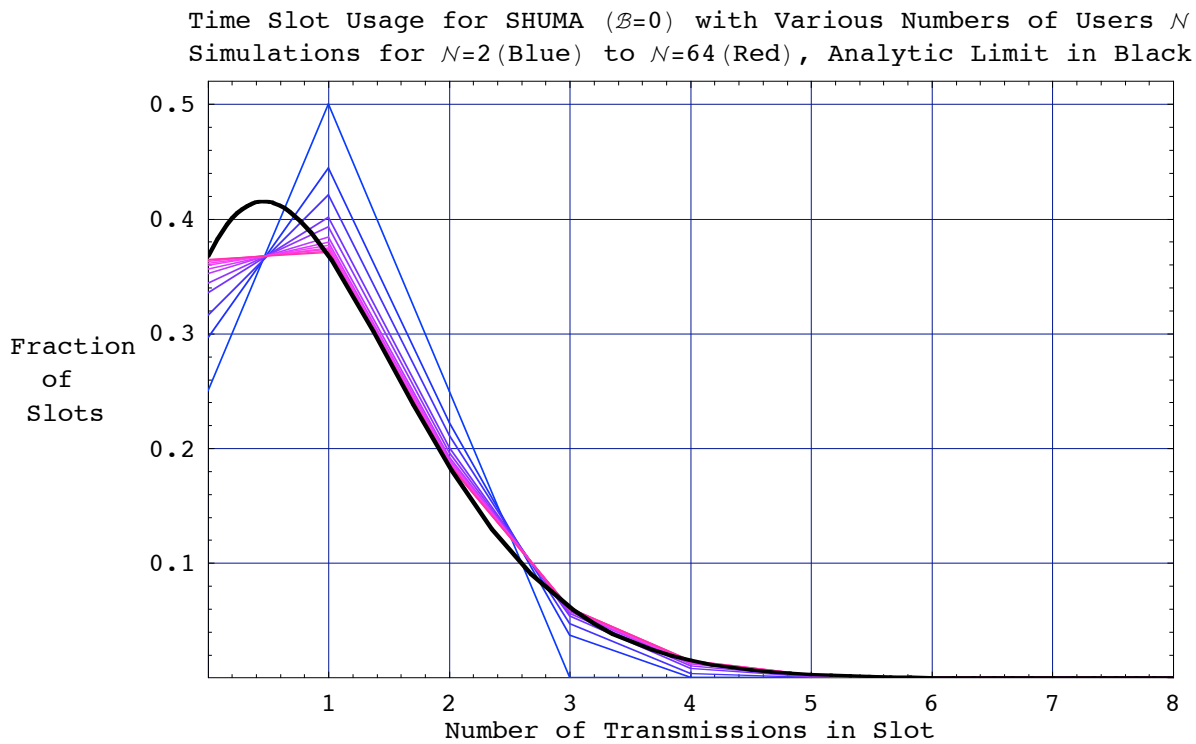


Figure 4. Time slot usage for SHUMA for various $\mathcal{N}$. Simulation results show time slot usage as the number of users ranges from 2 (blue) to 64 (red). The analytic result shown in black predicts time slot usage when the number of SHUMA participants is large. Each simulation curve summarizes behavior observed over the course of $5 \cdot 10^6$ time slots, and straight lines connect sample points. Theoretical predictions were converted from discrete to continuous form by replacing $\mathbb{k}!$ with `Gamma[`$\mathbb{k}+1$`]` in *Out[11]*.

As we saw in Figure 3, the fraction of time slots with a single transmission is maximized under CA only when the number of users equals the number of time slots in an access interval, and this typically limits CA to applications where the number of users is constrained in some way (i.e., it may be known that messages need to be exchanged only among a group of "nearest neighbors"). On the other hand, the time slot usage shown in Figure 4 suggests that SHUMA can provide the best performance that CA can provide even when the number of users is not constrained.

4. Average Receive Intervals

Roughly stated, Average Receive Interval (ARI) is a measure of how often transmissions from one unit reach another when there is some specific number of intervening participants. ARI is relatively simple to calculate for DA because collisions do not occur; thus, for example, if a user is allocated σ time slots per frame, then we can say that the ARI from that user has the constant value $12/\sigma$ seconds for any receiver. Under CA and SHUMA, ARI is a function of the number of intervening participants because those intervening participants are capable of transmitting and thus blocking a message.

■ 4.1 CA Average Receive Interval

Under CA, we know that each unit will transmit once during each access interval, and so we can consider only those slots in which the transmission occurs. Suppose that the number of access intervals per frame is denoted by α, and the number of slots per frame is given by s, and suppose that between the transmitter and receiver under consideration, there are i intervening (potentially transmitting) terminals. For a transmission to be received when i intervening units are present, it is necessary that all of these intervening units be silent, which happens with probability $(1-\alpha/s)^i$, and will occur once per $1/(1-\alpha/s)^i$ access intervals on the average. Since access intervals occur at a rate of $12/\alpha$ per second, the average interval, in seconds, between successful transmissions between units that are separated by i units is

$$\mathrm{ARI}_{\mathrm{CA}}(i) = \left(\tfrac{12}{\alpha}\right)\left(\tfrac{1}{1-\alpha/s}\right)^i \tag{5}$$

Note that regardless of the number of allocated time slots, and regardless of the relative positions of the platforms, $\mathrm{ARI}_{\mathrm{CA}}$ cannot become less than $12/\alpha$.

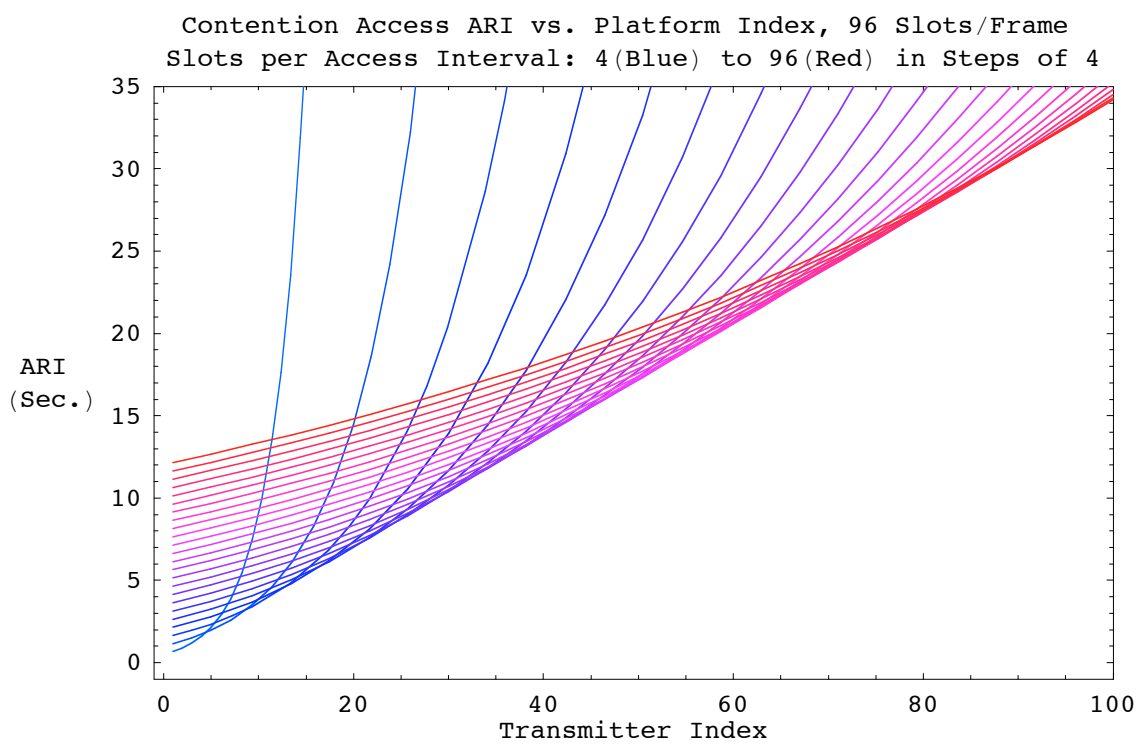


Figure 5. Average Receive Intervals under Contention Access with 96 slots/frame. Slots per access interval range from 4 to 96 in steps of 4, and thus for the receiver closest to the transmitter access interval length ranges from 1/2 second to 12 seconds. ARI makes sense only at integer values of transmitter index, but is plotted continuously for clarity.

The ARI performance of CA with 96 slots/frame is shown in Figure 5 for a range of access interval sizes. It should be kept in mind that the horizontal axis represents the order of transmitters from a single receiver (as shown, for example, by the indices in Figure 1), and that the curves are parameterized in terms of the number of slots per access interval, as opposed, say, to the number of access intervals per frame.

Figure 5 illustrates that short access intervals allow transmissions to be made more frequently, but lead to collisions (i.e., high ARI) even when there are few users. In contrast, long access intervals force transmissions to be

made less frequently, but there are more time slots per access interval and so collisions do not predominate until many more participants are present. Significantly, because the length of an access interval is fixed at network design time, performance follows the same fixed curve regardless of the number of participants.

■ 4.2 SHUMA Average Receive Intervals

With s denoting the number of slots per frame, a transmission across i intervening units will be successful if the transmitter under consideration does in fact transmit while the i intervening units do not transmit. This occurs with a probability of $p(1-p)^i$, which will occur on the average about once out of $1/(p(1-p)^i)$ trials. Since there will be $12/s$ seconds between trials, the time between successful receptions under SHUMA is given by the following.

$$\mathrm{ARI}_{\mathrm{SHUMA}}(i) = \left(\frac{12}{s}\right)\frac{1}{p(1-p)^i}\ , \quad \text{with } p = 1-(1-1/\mathcal{N})^{\mathcal{B}+1} \tag{6}$$

A key characteristic of SHUMA is that ARI performance changes as the number of participants changes; that is, in contrast to behavior under CA, the ARI experienced by any particular unit changes as other participants come and go. The figure below gives SHUMA ARI performance with $\mathcal{B} = 0$.

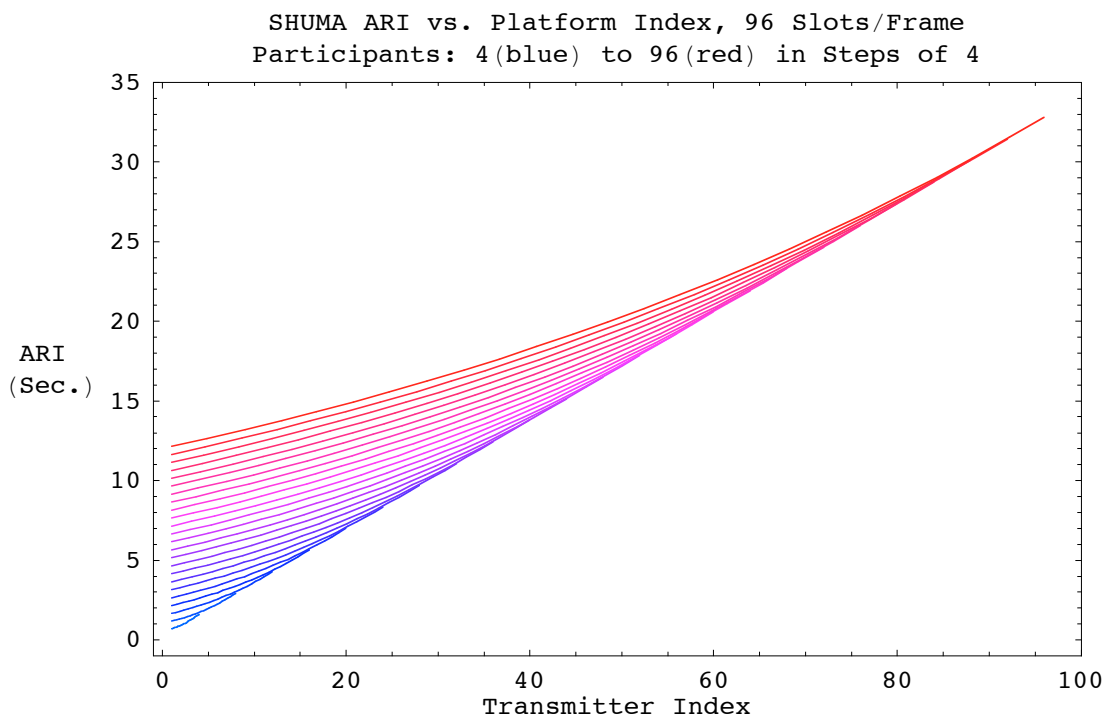


Figure 6. Average Receive Intervals under SHUMA with 96 slots/frame. The number of participants ranges from 4 to 96. This figure illustrates how ARI performance under SHUMA changes as a function of the number of participants sharing the time slot pool. Note that these curves are not fixed at network design time, but rather reflect the number of participants in the network.

In an idealized case, no collisions would occur and Figure 6 would consist of a series of horizontal lines at values of $12\,\mathcal{N}/96 = \mathcal{N}/8$, with each line extending horizontally from 1 to $\mathcal{N}$. Though no protocol can exactly match this ideal performance, Figure 6 indicates that SHUMA provides a reasonable approximation. The ARI performance that a SHUMA participant experiences is a function not only of its index, but also on the total number of participants. More specifically, Figure 5 shows that when the number of participants increases, *all* participants transmit less often, which allows participants of higher index to make successful transmissions more frequently. This figure thus illustrates how SHUMA participants "think globally and act locally."

It turns out that a simple and meaningful relationship holds between the ARI expressions for SHUMA and CA.

ARI values under SHUMA equal those under CA when the SHUMA probability of transmission equals the reciprocal of the number of slots in an access interval. Here, α is the number of access intervals per frame, and s is the number of slots per frame.

In[12]:= $$\frac{(12/\alpha)}{(1-\alpha/s)^{i-1}} == \frac{(12/s)}{p\,(1-p)^{i-1}} \;/.\; p \to \alpha/s$$

Out[12]= True

The term s/α is the number of CA time slots in an access interval. Thus, this relationship simply states that the ARI experienced under SHUMA is the same as that experienced under CA when the SHUMA probability of transmission is inversely proportional to the number of CA slots in each access interval. If we fix $\mathcal{B}$ at zero, the SHUMA probability of transmission becomes $p = 1/\mathcal{N}$, and the relationship $p = \alpha/s$ implies that the ARIs for SHUMA and CA will be the same when $p = 1/\mathcal{N} = \alpha/s$, or $\mathcal{N} = s/\alpha$. That is, in the important special case of SHUMA with $\mathcal{B} = 0$, SHUMA ARI performance is identical to that of the "optimal" CA case of one time slot per access interval per user. Note that this performance is achieved under CA only when the number of users equals the number of time slots in a CA access interval. Even if the number of users does happen to be fixed for the full duration of a mission, it is not necessary to know this number at network design time under SHUMA

5. Channel Access Latency

A terminal under SHUMA is never forced to make a transmission: in anticipation of an upcoming SHUMA time slot, a terminal essentially generates a random number, and makes a transmission only if that random number falls within a certain specified range. Thus it's possible that long runs of SHUMA time slots may occur during which a particular terminal makes no transmissions. In this section we characterize the length and frequency of intervals between transmissions under SHUMA, and discuss their potential significance.

Assuming that the terminal has data to send, SHUMA transmissions can be modeled as Bernoulli trials with a probability of "success" given by Equation (1). The interval between transmissions will then follow a geometric distribution. That is, after a transmission has occurred, the probability that k time slots will pass before the next transmission is

$$\tau(k) = p(1-p)^k, \text{ with } p = 1 - (1 - 1/\mathcal{N})^{\mathcal{B}+1}. \tag{7}$$

Because $p < 1$, the probability of an interval of length k decreases exponentially as k increases. It should be kept in mind that $\mathcal{B}$ is generally set to a value other than zero only when transmissions are expected to be bursty, so that when $\mathcal{B} \neq 0$ these trials are not necessarily independent.

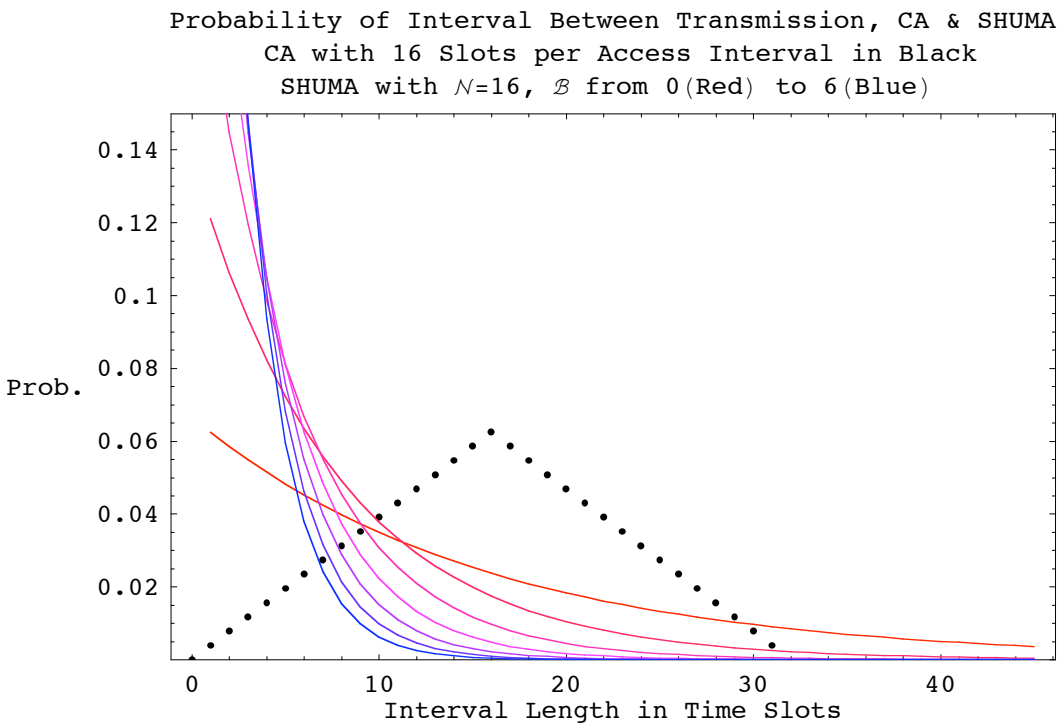


Figure 7. Probabilities associated with intervals between transmissions for contention access and for SHUMA. Note that the count of slots between transmissions includes exactly one (either one) of the bounding slots. Thus with an access interval of 16 time slots, the average interval between transmissions is 16 slots. Note that the SHUMA probabilities are discrete, like CA, but are shown here as continuous for clarity.

In Figure 7 the interval between transmissions is taken to be the number of intervening time slots, plus one of the bounding slots (where a "bounding" slot is one in which a transmission occurs on one or the other end of the interval). For example, if a CA transmission occurs in the very last slot of an access interval and the very next slot of the next access interval, then we count this as an interval between transmissions of one. To obtain the probability density function for intervals between CA transmissions, two uniform density functions were convolved.

It is possible to get simple closed form expressions for the probability that the number of time slots between transmissions will exceed a particular interval. We choose to measure the interval between transmissions in units of $\mathcal{N}$ time slots, which leads to a simple expression when $\mathcal{N}$ is large.

The probability that there will be $k \cdot \mathcal{N}$ or more time slots between SHUMA transmissions can be expressed as follows.

In[13]:= $\sum_{\mathbb{k}=k\,\mathcal{N}}^{\infty} p\,(1-p)^{\mathbb{k}}$ `/. p → 1 / 𝒩 + (1 - 1 / 𝒩) (1 - (1 - 1 / 𝒩)^ℬ) // FullSimplify`

Out[13]= $\left(\left(\frac{-1+\mathcal{N}}{\mathcal{N}}\right)^{1+\mathcal{B}}\right)^{k\,\mathcal{N}}$

This expression approaches a simple exponential as $\mathcal{N}$ increases.

```
In[14]:= Limit[%, 𝒩 → ∞]
```

Out[14]= $e^{-(1+\mathcal{B})k}$

This result tells us that when $\mathcal{N}$ is large, a terminal with $\mathcal{B}$=0 will have an interval between transmissions of $\mathcal{N}$ with probability $e^{-1} \approx 0.37$, an interval between transmissions of $2 \cdot \mathcal{N}$ with probability $e^{-2} \approx 0.14$, an interval between transmissions of $3 \cdot \mathcal{N}$ with probability of $e^{-3} \approx 0.050$, and so on.

Thus, although it is possible that long intervals can occur between SHUMA transmissions, the probability that an interval of a particular length will occur decreases exponentially as the interval length increases.

6. Summary & Conclusions

This paper has shown that the behavior of a SHUMA network can be viewed as an adaptive, scalable form of CA. For the important special case in which the probability of transmission is equal to $p = 1/\mathcal{N}$ (that is, when $\mathcal{B} = 0$), SHUMA operates much like a CA network with an access interval that is constantly adjusted to contain $\mathcal{N}$ slots. Figure 4 illustrated that, regardless of the number of users $\mathcal{N}$, the fraction of time slots in which a single transmission occurs is better than, or at worst equal to, that of CA. It was also shown (in Figure 6) that SHUMA ARI performance is maintained "below the knee" of the corresponding CA curve. If desired, the adaptive nature of SHUMA can be restricted or eliminated (by bounding the parameter $\mathcal{N}$ above and/or below) to mimic the constant transmission rate of CA and its consequent properties.

Although SHUMA can be viewed in terms of CA, the mechanism which drives SHUMA behavior is fundamentally different than that of any static protocol, and is in fact different than that of most adaptive protocols. SHUMA is distinguished by its use of situational awareness data, and in particular by the data gathered through normal exchange of PPLIs. This information is available within the terminal whether or not it is used by the channel access protocol, and it thus serves a dual purpose: it provides valuable information to the Link-16 user, and it allows SHUMA to regulate transmissions so that large fractions of time slots contain single transmissions. Because the adaptive behavior of SHUMA is achieved with no channel capacity being consumed by overhead, update rates under SHUMA are limited only by the capacity of the Link-16 channel, regardless of the number of users.

The close relationship between the SHUMA concept and the Link-16 environment deserves special mention. SHUMA exploits one of the most characteristic applications of Link-16, situational awareness, to satisfy one of its most significant requirements, namely that transmissions receive the widest dissemination possible, and the resulting performance is a good complement to the performance characteristics of constant transmission rate protocols. SHUMA is not an *ad-hoc* collection of mechanisms appropriate for only a limited number of specific operational scenarios.

Finally, of course, SHUMAs characteristics are significant only because they provide a way to meet current and projected needs of users. As the number of Link-16 users in theatre increases, and as the flexibility they require grows, the significance of SHUMA is expected likewise to increase

References

1. *Combined/Joint Network ACCO0004A Network Description*. USMC Network Design Facility, Marine Corps Tactical Systems Support Activity, Camp Pendleton, CA, March 7, 2003.

2. *Dynamic Reconfiguration of Link-16: Office of Naval Research Project FY01 Summary*. SPAWAR Systems Center, San Diego, September 2001.
3. *Understanding Link-16: A Guidebook for New Users*, Navy Center for Tactical Systems Interoperability (NCTSI), September 2001.
4. Wolfram, Stephen. The *Mathematica* Book, 5th Ed. Wolfram Media, 2003.

Appendix A. Code For Figures

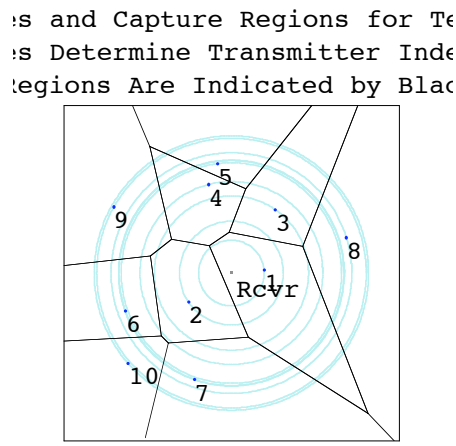
The code below was used to generate the figures that appear in the body of this document. To save space, the size of the figures below has been greatly reduced. As a consequence, some of the figure titles have been cropped.

■ A1. The Capture Effect (Figure 1)

The integers 8, 7 and 5 are good random number seeds for making this figure most effective.

```
In[1]:= Needs["DiscreteMath`ComputationalGeometry`"];
        Needs["Graphics`Colors`"];

In[3]:= Remove[r, rXmit, rRcvr, α, β, χ];
        SeedRandom[8];
        r = Table[{Random[Real, {-1, 1}], Random[Real, {-1, 1}]}, {30}];
        rXmit = Sort[Take[r, 10], #1.#1 < #2.#2 &];

In[7]:= {PaleTurquoise, Circle[{0, 0}, √#.#]} & /@ rXmit;
        {Blue, Point[#]} & /@ rXmit;
        α = Graphics[Join[{PointSize[0.01]}, %%, %,
            {Gray, Rectangle[{-0.01, -0.01}, {0.01, 0.01}]}]];

In[10]:= β = DiagramPlot[rXmit, LabelPoints → True,
             DisplayFunction → Identity] /. PointSize[_] → PointSize[0] /.
           Text[a_, b_] :→ Text[a, b, {-1, 1}] /.
          Thickness[_] → Thickness[0.0005];

In[11]:= χ = Graphics[Text["Rcvr", {0, 0}, {-1, 1}]];
```

In[12]:=
```
Show[α, β, χ, PlotRange → {{-1.41, 1.41}, {-1.41, 1.41}},
  AspectRatio → 1, Frame → True, FrameTicks → False,
  PlotLabel → "Terminal Indices and Capture Regions for Ten
     Transmitters\n\tBlue Circles Determine Transmitter Index
     wrt Rcvr\n\t  Capture Regions Are Indicated by Black Lines",
  DisplayFunction → $DisplayFunction];
```



■ A2. Effects of Errors in Estimate of $\mathcal{N}$ (Figure 2)

In[17]:= $\left\{p_{act} \to \frac{1}{\mathcal{N}},\ p_{est} \to \frac{1}{(1+\varepsilon)\,\mathcal{N}}\right\};$

In[18]:= $\frac{\mathcal{N}\, p_{act}\,(1-p_{act})^{\mathcal{N}-1} - \mathcal{N}\, p_{est}\,(1-p_{est})^{\mathcal{N}-1}}{\mathcal{N}\, p_{act}\,(1-p_{act})^{\mathcal{N}-1}}$ /. % // FullSimplify

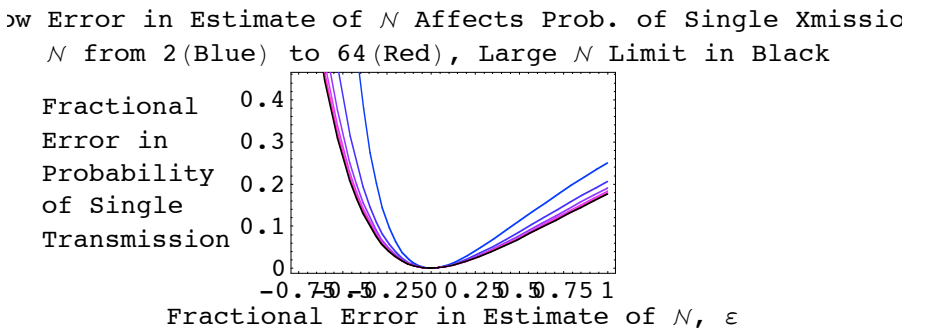
Out[18]= $1 - \frac{(-1+\mathcal{N})^{1-\mathcal{N}}\,\mathcal{N}^{\mathcal{N}}\,\left(1-\frac{1}{\mathcal{N}\,(1+\varepsilon)}\right)^{\mathcal{N}}}{-1+\mathcal{N}+\mathcal{N}\,\varepsilon}$

In[19]:=
```
Limit[%, 𝒩 → ∞] // FullSimplify
```

Out[19]= $1 - \frac{e^{\frac{\varepsilon}{1+\varepsilon}}}{1+\varepsilon}$

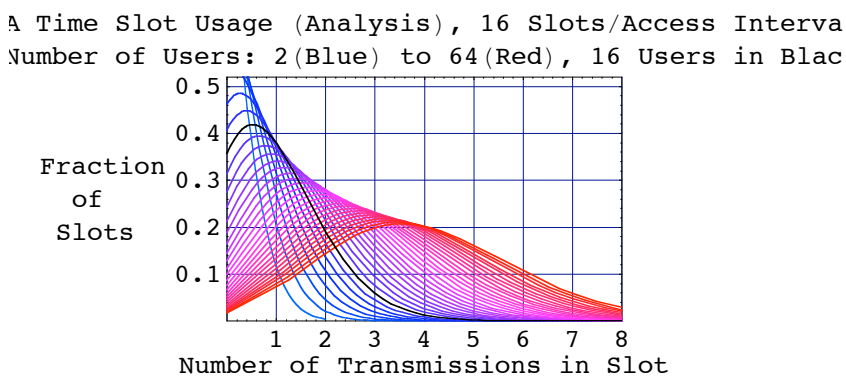
In[20]:=
```
Plot[Evaluate[%% /. 𝒩 → {2, 4, 8, 16, 32, 64}],
  {ε, -0.75, 1}, PlotStyle → Table[Hue[i], {i, 0.65, 1, 0.05}],
  DisplayFunction → Identity];
Plot[1 - Exp[ε / (1 + ε)] / (1 + ε), {ε, -0.75, 1},
  DisplayFunction → Identity];
```

In[22]:=
```
Show[%%, %, PlotLabel →
   "How Error in Estimate of 𝒩 Affects Prob. of Single Xmission\n\
    t𝒩 from 2(Blue) to 64(Red), Large 𝒩 Limit in Black",
  FrameLabel → {"Fractional Error in Estimate of 𝒩, ε",
    "Fractional\nError in\nProbability\nof Single\nTransmission"},
  DisplayFunction → $DisplayFunction];
```



■ A3. Time Slot Usage Under CA, Analytic Results (Figure 3)

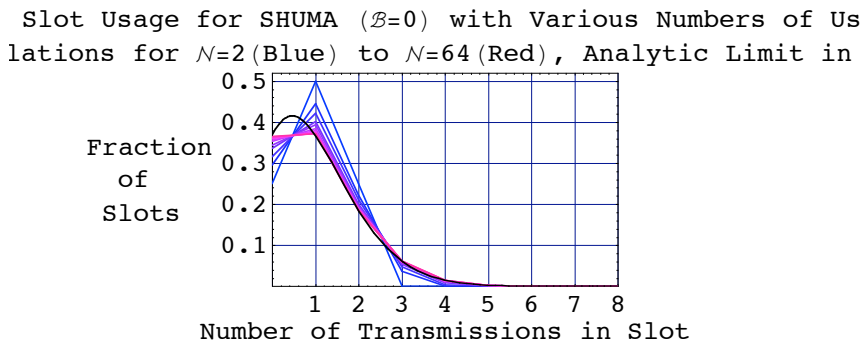
In[45]:=
```
Plot[Evaluate[
     Binomial[𝒩, 𝕜] (1 / s)^𝕜 (1 - 1 / s)^(𝒩-𝕜) /. {s → 16., 𝒩 → #}], {𝕜, 0, #},
    PlotStyle → Hue[0.6 + # / 160], DisplayFunction → Identity] & /@
  DeleteCases[Range[2, 64, 2], 16];
Append[%, Plot[Evaluate[
    Binomial[𝒩, 𝕜] (1 / s)^𝕜 (1 - 1 / s)^(𝒩-𝕜) /. {s → 16., 𝒩 → 16}], {𝕜, 0, 16},
   PlotStyle → Thickness[0.004], DisplayFunction → Identity]];
Show[%, PlotRange → {{0, 8}, {0, 0.52}},
  GridLines → {Range[0, 8], Range[0.1, 0.5, 0.1]},
  DisplayFunction → $DisplayFunction,
  FrameLabel → {"Number of Transmissions in Slot",
    "Fraction\n  of\n Slots"}, PlotLabel →
   "CA Time Slot Usage (Analysis), 16 Slots/Access Interval\n
     Number of Users: 2(Blue) to 64(Red), 16 Users in Black"];
```



■ A4. Time Slot Usage Under SHUMA, Simulation & Analysis (Figure 4)

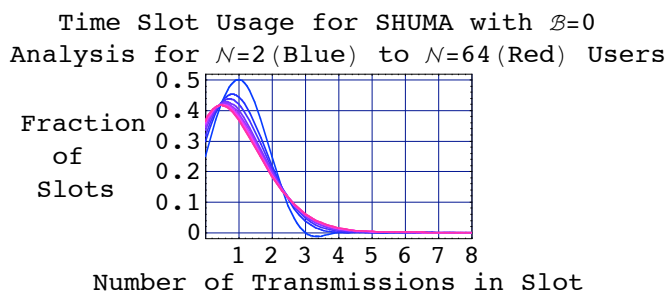
In[1]:=
```
Remove[shumaTSU];
shumaTSU[numOfUsers_Integer] :=
 Module[{t1, t2, t3},
  t1 = Table[
    Random[Integer, {1, numOfUsers}], {5 10^6}, {numOfUsers}];
  t2 = Count[#, 1] & /@ t1;
  t3 = {#, Count[t2, #] / Plus @@ t2} & /@ Range[0, 8];
  Return[t3]
 ]
```

```
In[3]:= ListPlot[shumaTSU[#], PlotJoined → True,
          PlotStyle → Hue[0.6 + Log[2, #] / 20], (*,#/160*)
          DisplayFunction → Identity] & /@
        {2, 3, 4, 6, 8, 12, 16, 22, 28, 36, 48, 64};
      Plot[1 / (ⅇ Gamma[𝕜 + 1]), {𝕜, 0, 8}, PlotStyle → Thickness[0.004],
        DisplayFunction → Identity];
      Show[Flatten[{%%, %}], PlotRange → {{0, 8}, {0, 0.52}},
        GridLines → {Range[0, 8], Range[0.1, 0.5, 0.1]}, FrameLabel →
         {"Number of Transmissions in Slot", "Fraction\n  of\n Slots"},
        PlotLabel → "Time Slot Usage for SHUMA (ℬ=0) with
           Various Numbers of Users 𝒩\nSimulations for 𝒩=2
           (Blue) to 𝒩=64(Red), Analytic Limit in Black",
        DisplayFunction → $DisplayFunction];
```



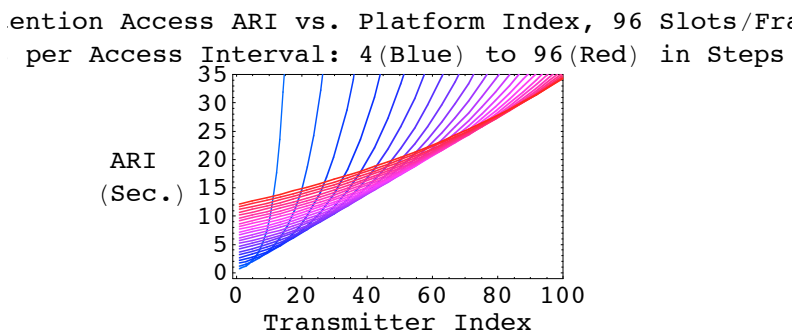
In[13]:=
$$\frac{\left(1-\left(\frac{-1+\mathcal{N}}{\mathcal{N}}\right)^{1+\mathcal{B}}\right)^{\mathbb{k}}\left(\left(\frac{-1+\mathcal{N}}{\mathcal{N}}\right)^{1+\mathcal{B}}\right)^{\mathcal{N}-\mathbb{k}}\text{Gamma}[\mathcal{N}+1]}{\text{Gamma}[\mathcal{N}-\mathbb{k}+1]\,\text{Gamma}[\mathbb{k}+1]}\ /.\ \mathcal{B}\to 0;$$

```
Plot[Evaluate[% /. 𝒩 → #], {𝕜, 0, 8}, PlotStyle →
     Hue[0.6 + Log[2, #] / 20], DisplayFunction → Identity] & /@
  {2, 3, 4, 6, 8, 12, 16, 22, 28, 36, 48, 64};
Show[%, DisplayFunction → $DisplayFunction,
  PlotRange → {{0, 8}, {-0.02, 0.52}},
  GridLines → {Range[0, 8], Range[0, 0.5, 0.1]}, FrameLabel →
   {"Number of Transmissions in Slot", "Fraction\n  of\n Slots"},
  PlotLabel → "   Time Slot Usage for SHUMA with ℬ=0
     \nAnalysis for 𝒩=2(Blue) to 𝒩=64(Red) Users",
  DisplayFunction → $DisplayFunction];
```



■ A5. ARI Under CA (Figure 5)

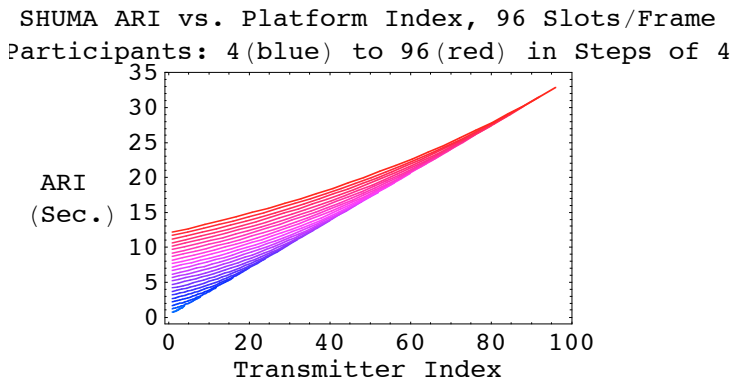
```
In[185]:= Table[{(sPerA / 8) (1 - 1 / sPerA)^-i, sPerA}, {sPerA, 4, 96, 4}];
Plot[#[[1]], {i, 1, 100}, PlotStyle → Hue[(150 + #[[2]]) / 250],
    FrameLabel → {"Transmitter Index", " ARI\n(Sec.)"},
    PlotLabel → " Contention Access ARI vs. Platform Index, 96
       Slots/Frame\nSlots per Access Interval: 4(Blue) to 96
       (Red) in Steps of 4", DisplayFunction → Identity] & /@ %;
Show[%, PlotRange → {{-1, 100}, {-1, 35}},
  DisplayFunction → $DisplayFunction];
```



■ A6. ARI Under SHUMA (Figure 6)

$$\mathrm{ARI}_{\mathrm{SHUMA}}(i) = \left(\tfrac{12}{s}\right) \tfrac{1}{p(1-p)^i}\ , \quad \text{with } p = 1 - (1 - 1/\mathcal{N})^{\mathcal{B}+1} \approx (\mathcal{B}+1)/\mathcal{N} \tag{8}$$

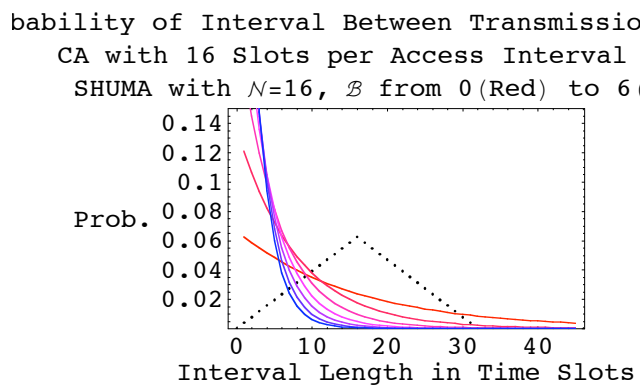
```
In[39]:= Plot[Evaluate[(12 / s) (#) (1 - 1 / #)^-i /. s → 96],
            {i, 1, #}, PlotStyle → Hue[(150 + #) / 250],
            DisplayFunction → Identity] & /@ Range[4, 96, 4];
         Show[%, PlotRange → {{-1, 100}, {-1, 35}},
           DisplayFunction → $DisplayFunction,
           FrameLabel → {"Transmitter Index", " ARI\n(Sec.)"},
           PlotLabel → " SHUMA ARI vs. Platform Index, 96 Slots/Frame
              \nParticipants: 4(blue) to 96(red) in Steps of 4"];
```



■ A7. Channel Access Latency & Related Notes (Figure 7)

The number of SHUMA time slots between transmissions follows a geometric distribution. We convolve two uniform distributions (using the associated polynomial/generating function) to get the probability density of the interval between transmissions for CA.

```
In[33]:= Remove[p, pShuma, pCa];
SetOptions[ListPlot, Frame → True, Axes → False];
x^16 (Sum[1/16 x^t, {t, 1, 16}]) (Sum[1/16 x^-t, {t, 1, 16}]) // Expand;
CoefficientList[%, x];
Transpose[{Range[0, Length[%] - 1], %}];
pCa = ListPlot[%, DisplayFunction → Identity];
p[𝒩_, ℬ_] := 1 / 𝒩 + (1 - 1 / 𝒩) (1 - (1 - 1 / 𝒩)^ℬ);
pShuma = ListPlot[
     Table[{t, p[16, #] (1 - p[16, #])^(t-1)}, {t, 1, 45}],
     PlotStyle → Hue[1 - # / 18], PlotJoined → True,
     DisplayFunction → Identity] & /@ Range[0, 6];
Show[Flatten[{pCa, %}], DisplayFunction → $DisplayFunction,
  FrameLabel → {"Interval Length in Time Slots", "Prob."},
  PlotRange → {0, 0.15},
  PlotLabel → "Probability of Interval Between Transmission,
     CA & SHUMA\n\t  CA with 16 Slots per Access Interval in
     Black\n\t   SHUMA with 𝒩=16, ℬ from 0(Red) to 6(Blue)"];
```



■ A8. The Initialization File *init.m*

The *Mathematica* initialization file `init.m` used to create this document contains the following lines.

```
SetOptions[Plot, Frame→True, Axes→False, RotateLabel→False];
SetOptions[ListPlot, Frame→True, Axes→False, RotateLabel→False];
SetOptions[TableForm, TableDirections→{Row, Column}];
```