F2

Uplink Multiple Access Techniques for Satellite Communication Systems

by

Christopher J. Karpinsky

B.S. Electrical Engineering
Rochester Institute of Technology, 1996

Submitted to the department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 1998

Signature of Author:

Christopher J. Karpinsky
Department of Electrical Engineering and Computer Science
May 15, 1998

Certified By:

Dr. Vincent W.S. Chan
Head, Communications and Information Technology Division
MIT Lincoln Laboratory
Thesis Supervisor

Accepted By:

Arthur C. Smith
Chair, Committee on Graduate Students

Uplink Multiple Access Techniques for Satellite Communication Systems

by

Christopher J. Karpinsky

Submitted to the Department of Electrical Engineering and Computer Science on June, 1998, in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering and Computer Science

Abstract

In this thesis uplink multiple access communication techniques for satellite data communication systems are considered. Background is offered to demonstrate the need for the application of efficient multiple access techniques to systems characterized by a large number of bursty unscheduled users. An overview of multiple access transmission techniques applicable to the satellite channel is presented with focus on delay versus throughput characteristics. Satellite uplink architecture comparisons are made showing the inefficiencies of a channelized architecture when applied to bursty data traffic. A dynamically-assigned uplink architecture is presented which offers superior delay versus throughput performance over that achievable with a channelized uplink. Additional performance gain of the dynamically-assigned uplink is realized while considering reconfigurable uplink architectures which dynamically adapt to changing system loading. Both uplink architectures are analyzed with two distinctly different communication requirements. The first communication service investigated is that of large message transfer from bursty sources, modeling the performance experienced with file transfer. The second communication service considered follows from the client-server transmission model. Here we are interested in affecting efficient uplink communication from many user client applications to a single server, a mode of communication resembling the hypertext transport functionality of the World Wide Web. The superiority of the dynamically-assigned uplink architecture is demonstrated for both of these communication requirements. Additional techniques of performance evaluation are introduced with an overview of the Markov modulated Poisson process (MMPP). The MMPP can be used for modeling packet arrival processes from general data sources as well as approximating complex packet arrival processes such as those experienced with local area network traffic. Finally, we consider the generalization of multiple access to vector space. This formulation offers an intuitive visual framework for the consideration of multiple access coding, the process of situating multiple user transmissions in a high dimensional space while minimizing multiple access interference between users. As an example of multiple access coding, we discuss a generalization of the Slotted ALOHA transmission technique which allows multiple transmissions per time slot.

Thesis Advisor: Dr. Vincent W.S. Chan
Head, Communications and Information Technology Division
MIT Lincoln Laboratory

Contents

Chapter 1

Satellite Multiple Access Communications

1.1 Introduction

An interesting article was published in *Wireless World* in October of 1945 entitled "*Extra-Terrestrial Relays*". This article described the radical concept of placing a device into orbit around the earth at a distance of 42,000 km, an orbit that is synchronous with the rotation of the earth. This device, appearing to be stationary in the sky, would then be available for the broadcast of information to a large geographical area. Arthur C. Clarke was the engineer who held this vision of offering broadcast services from space. Clarke's system consisted of three such devices, interconnected with either radio or optical links, covering the entire globe. The idea of satellite communications was born.

We have witnessed the fruition of Clarke's idea over the past thirty years. Satellite communication is used regularly for broadcasting and telephony trunking. Transponding satellites continually relay streams of analog information around the world by what is known as *circuit switching*. With user demand shifting in recent years towards digital transmission we see satellite communication equipment becoming smaller and cheaper as intelligent forms of modulation and coding are used. Satellite communication requirements have moved from large, statistically multiplexed trunking gateways with large apertures to small hand-held terminals with large variation of instantaneous communication requirements.

Intelligent methods of sharing resources becomes more and more important as needs shift from quasi-statically-assigned circuit-switched analog services to rapidly varying *packet-switched* digital communications. *Multiple access* is the study of how a collection of users share a single communication resource in an efficient manner. In terms of the open systems interconnection (OSI) model shown in Fig. 1.1, multiple access appears as the medium access control (MAC) layer above the bottom physical layer. The MAC layer coordinates the use of the shared resource and dictates the discipline all users must follow to affect efficient communication.

Multiple access techniques have existed for quite a while allowing multiple streams to be transmitted simultaneously through a satellite transponder. Packet-switched communications, however, place higher demands on the MAC layer as user data requirements change. Additionally, with the move towards personal computer communication systems, the data traffic offered to the MAC layer is not an aggregation of multiple streams and hence tends to be more *bursty* in nature. For this type of traffic we wish to investigate efficient *random access* schemes which allow bursty packet-switched systems to access the communications medium with low delay.

Our intent in this chapter is to investigate efficient multiple access techniques as applied to the problem of sending bursty packet-switched data up to a communication satellite. The specific application to satellite channels takes into account the large propagation delay associated with satellite communications. With an understanding of known multiple access schemes for transmitting packet data, we will be prepared to analyze the performance of two distinctly

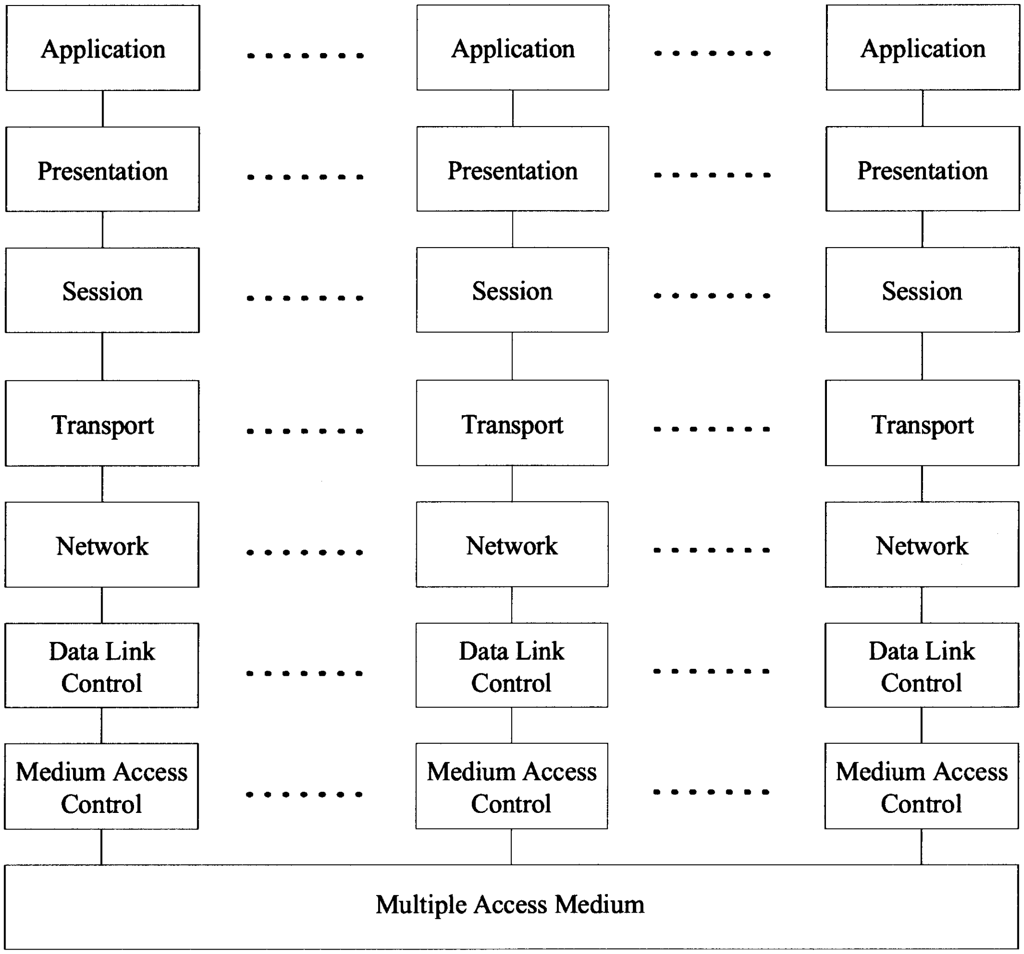


Figure 1.1: The open systems interconnection (OSI) model defines a layered model to help with the conceptual understanding of network architecture. Note the position of the medium access control layer and its function as the interface to the shared communication resources.

different communication architectures in the next chapter. Further elaboration of the topics covered in this Thesis is offered at the end of this chapter, motivated by the concepts which follow.

1.2 Metrics for Evaluating Multiple Access Techniques

The two fundamental resources available for information transmission are time and frequency. A multiple access scheme strives to coordinate the use of these two resources in an efficient way while minimizing interference between users. Determination of the "optimal" multiple access technique for a given system is a difficult task, assuming such an effort is even possible. It is important to begin by identifying the various measures of performance that are available while evaluating various multiple access techniques. The following is a brief summary of the major factors and parameters one uses during evaluation.

Orthogonality In the above description of the goals of all multiple access schemes, the concept of minimizing interference between users entails separating users in the time and/or frequency dimensions. A scheme which provides 100% isolation between users is referred to as an *orthogonal* multiple access technique. Schemes with weaker isolation between users are referred to as *quasi-orthogonal.* Additionally, schemes that offer no isolation between users are known as *random* access or *contention-mode* multiple access.

Throughput Again, the above description calls for the *efficient* use of time and frequency. This efficiency achieved by the multiple access technique is quantified by its *maximum throughput*, or maximum achievable utilization of available channel capacity.

Delay The general trade-off for maximizing throughput is the necessary increase of message transmission time or *message delay.* The most common evaluation of a multiple access scheme is its delay-throughput characteristic. For example, we will see that TDMA is able to operate with a maximal throughput of unity, but such operation is accompanied by infinite message delay.

Stability A concern for random access schemes (where collisions between user transmissions frequently occur) is the development of a contention resolution algorithm and, consequently, this algorithm's *stability.* Contention between users for time and/or frequency resources must be handled in such a way that the operating point on the delay-throughput characteristic is well-behaved.

Fairness Not only must the multiple access scheme isolate user transmissions in such a way to ensure high throughput and low delay, the scheme must also do so in a *fair* fashion, that is, all users must have equal access rights to the communication medium. Fairness is an issue in systems with heterogeneous users as well as homogeneous users. The user with small data requirements should not be blocked by a user with large data requirements. Additionally, in a homogenous system, one user cannot hold the channel active for long periods of time, disregarding other similar users' requests.

Duty Cycle A third fundamental resource beyond time and frequency, albeit a bit less restrictive, is average transmit power. The transmit *duty cycle* of a multiple access scheme is the fraction of time the transmitter is active. A low duty cycle technique can offer much higher instantaneous transmit power while keeping average power the same as a full duty cycle technique. This characteristic is important when considering power limitations imposed by battery storage or solar generation.

1.3 Orthogonal Multiple Access Techniques

We begin this section by describing the distinction between multiplexing and multiple accessing. Conventional multiple access techniques are those in which orthogonality is provided between user transmissions thus affecting individual channels of communication. Frequency division and time division multiple access are described in addition to synchronous code division multiple access.

1.3.1 Multiplexing

Multiplexing is the foundation for achieving multiple access, it is the technique of creating multiple channels of communication from one larger channel. We review basic multiplexing before discussing common multiple access techniques.

Frequency Division Multiplexing

The most obvious technique for sharing communication resources between different services is to partition the frequency axis into disjoint frequency channels. This is frequency division multiplexing (FDM). Because each frequency channel is disjoint, we say FDM is an orthogonal technique for separating services.

Time Division Multiplexing

After frequency, the next resource to partition is time. For a given frequency channel, time division multiplexing (TDM) segments the time axis into disjoint time slots. Again, since the time slots are non-overlapping, we say TDM is an orthogonal multiplexing technique.

Code Division Multiplexing

A third technique for spitting a given communication channel into smaller subchannels is code division multiplexing (CDM). Time and frequency are still the only fundamental resources we have available, but instead of dividing each resource separately, CDM combines both TDM and FDM assigning unique orthogonal *codes* to each subchannel. CDM is further developed in Chapter 4 where we also show the mathematical equivalence of all three multiplexing techniques.

1.3.2 Frequency Division Multiple Access

Coordination of the frequency partitioning offered by FDM is the domain of frequency division multiple access (FDMA). FDMA is the discipline that assigns each user a unique channel, generally homogeneous, and disjoint from all other users. Orthogonality is inherited from FDM and we say the multiple access technique is orthogonal.

The expected delay for the transmission of a message via FDMA is trivial when we ignore the queuing delay associated with other messages (i.e. if we assume exceptional service). If a channel of capacity C [bits/sec] is shared in an FDMA fashion with N other users, a message of length L [bits] will experience an expected delay of

$$T_{\text{FDMA}} = \frac{NL}{C}. \tag{1.1}$$

1.3.3 Time Division Multiple Access

Likewise, coordination of the time partitioning created by TDM is the responsibility of time division multiple access (TDMA). As imagined, TDMA assigns each time slot to a single user thus providing orthogonality between users. The slots of each user are aggregated into a frame of duration T_f which forms a cyclical ring buffer for transmission.

We immediately imagine FDMA and TDMA to be equivalent multiple access techniques, that is, given a specific amount of frequency and time, the performance will be the same. This however is not the case. Following from above, we create a frame of duration $T_f = \frac{NL}{C}$ and recognize the expected message delay for TDMA as

$$T_{\text{TDMA}} = \frac{T_f}{2} + \frac{L}{C} \tag{1.2}$$

from which

$$\mathcal{T}_{\mathrm{TDMA}} = \mathcal{T}_{\mathrm{FDMA}} - \frac{L}{C}\left(\frac{N}{2} - 1\right) \tag{1.3}$$

follows. Thus, as N increases, we immediately see the advantage of TDMA over FDMA.

We next focus on considering delays associated with queuing. Even though TDMA, being an orthogonal multiple access scheme, provides isolation between users, we must consider the queuing delay associated with previous message transmission. Traditional TDMA occupying a channel of capacity C [bits/sec] is considered assuming each user with packets of length l_p [bits] is assigned a slot of duration $\tau = \frac{l_p}{C}$ [sec] in a frame of length $T_f = N\tau$ [sec]. The expected transmission time for a single user to send a message of length L [bits] is given by

$$X_{\mathrm{TDMA}} = \left(\left\lceil \frac{L}{l_p} \right\rceil - 1\right) T_f + \tau. \tag{1.4}$$

We must also consider, as messages arrive at an increasing rate, the expected wait due to queuing. From our assumptions of fixed-length messages, Poisson message arrivals, and infinite user buffering, an M/D/1 queue is formed with expected service rate $\mu = (X_{\mathrm{TDMA}})^{-1}$. The overall expected transmission delay is thus[1]

$$\mathcal{T}_{\mathrm{TDMA}}\left(\lambda_u, L, l_p, N, C\right) = \frac{T_f}{2} + \mathcal{Q}_{\mathrm{M/D/1}}\left(\lambda_u, (X_{\mathrm{TDMA}})^{-1}\right) + X_{\mathrm{TDMA}} + \mathcal{T}_R. \tag{1.5}$$

1.3.4 Code Division Multiple Access

For systems which employ spread spectrum in a synchronized fashion, we may separate users with their unique orthogonal psuedo-noise (PN) spreading sequence. It is important to note that only chip-synchronous (for direct sequence spreading) or hop-synchronous (for frequency hopped spreading) systems can be considered to be orthogonal multiple access schemes. The individual PN spreading sequences are chosen as the codes provide by CDM, uniquely identify each user's transmission, and also separate a particular user's transmission from all other transmissions. Because the code is the method for creating separate channels of communication, this transmission technique is known as *code division multiple access* (CDMA).

1.4 Random Access Protocols

We've seen above that it is a simple matter to divide the fundamental resources of time and frequency between multiple users. It might be intuitive to assign individual frequency channels to each user, but we now know that doing so is less efficient than assigning access time slots of some high rate channel to each individual user. Orthogonality implies the discipline that every user gets their share of transmission capacity, in some statically-assigned way.

An obvious deficiency with the above technique is revealed, however, when we think about non-constant stream sources accessing each orthogonal channel of TDMA, FDMA, or CDMA. Users with non-constant sources of information are classified as being ***bursty***, that is, their information transmission requirements vary instantaneously. A bursty source requires transmission resources whenever a message arrival occurs and is then silent until the next message arrival. The typical model for message arrivals of a bursty data source is the Poisson arrival model characterized by a mean arrival rate λ. Generally a source is labeled as being bursty if it has a high peak-to-average-traffic ratio. Because values of peak-to-average ratio of 1000:1 are not uncommon for computer-generated packet data, assigning a dedicated communication channel to a single bursty user immediately limits the efficient utilization of communication resources.

To affect more efficient use of communications resources, we must have a paradigm shift from the conventional orthogonal multiple access schemes described above to what are known as *random access protocols*. The original separation between multiplexing and multiple accessing

[1] Note our use of the functional queuing delay models $\mathcal{Q}_{\mathrm{M/D/k}}(\cdot)$ and $\mathcal{Q}_{\mathrm{M/D/1}}$ is described in Appendix A.

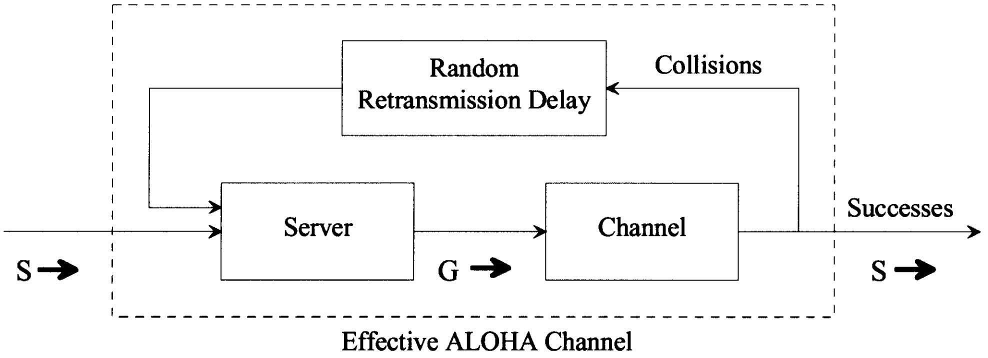


Figure 1.2: Logical description of the operation of an ALOHA channel.

described above for orthogonal transmission must be reconsidered. Random access schemes do just this–they combine the tasks of creating separate channels of communication with the appropriate assignment discipline. This resource access discipline allows for the possibility of two users' transmissions colliding with each other and hence demonstrate *contention* for access to the communication medium. Due to the possibility of collisions between user transmissions we consider random access schemes as being *non-orthogonal* multiple access techniques.

1.4.1 Pure ALOHA

The first communication system to implement a random access protocol was ALOHANET[1][2]. The system interconnected distributed computing resources through a satellite transponder. The random access protocol was named ALOHA and enforced a rather simple access discipline to the channel. If a computer had a message to transmit, it simply sent it. If, by the nature of the broadcast channel, the computer did not hear its own message relayed through the transponder, it would retransmit the message after a random delay. We refer to this ALOHA protocol as *pure ALOHA* to differentiate it from *slotted ALOHA* which will be discussed shortly.

To develop the expected delay associated with pure ALOHA we begin by assuming the aggregate arrival of packets from a large population of users is Poisson with parameter λ. Packets are of fixed-length τ [sec] and present an average input rate of $S = \lambda\tau$ [packets/packet length]. S is the offered traffic load to the pure ALOHA channel and is normalized such that $0 \leq S \leq 1$. The new packets must contend for access to the pure ALOHA channel. Due to contention, both newly-generated packets and retransmitted packets will offer an actual channel load of G [packets/packet length], as shown in Fig. 1.2.

Assuming a noiseless channel, a single packet is successfully transmitted if it neither overlaps with a current packet transmission nor suffers overlap by some other packet. We quantify this collision event by defining the notion of the *vulnerable period* of a packet. A packet of length τ transmitted at time t is vulnerable to overlap if any other user transmits a packet within $(t-\tau, t+\tau)$. We thus note the vulnerable period of a single packet to be 2τ [sec] or 2 [packet lengths]. Because the channel is experiencing traffic at the average rate of G [packets/packet length], the probability of k arrivals in an interval of t [packet lengths] is given by

$$\Pr[k,t] = \frac{(Gt)^k}{k!}\exp(-Gt). \tag{1.6}$$

Hence the probability of successful transmission of a single newly-generated packet is the probability of no other packet transmissions in an interval of 2 [packet lenghts]. Thus, for a newly generated packet, we have

$$\begin{aligned}\Pr[\text{success}] &= \Pr[k=0, t=2] \\ &= \exp(-2G) \end{aligned} \tag{1.7}$$

yielding overall channel throughput

$$S = G\exp(-2G) \tag{1.8}$$

with maximum achievable throughput

$$S_{\max} = \frac{1}{2e} \approx 0.184. \tag{1.9}$$

We see the performance of pure ALOHA is rather poor since, for example, an uplink with capacity of 100 [kbits/sec], although continually occupied, will only be able to pass approximately 18.4 [kbits/sec] of useful information. This poor throughput performance should be expected, however, since the rather liberal access discipline of transmit-at-will demonstrates the most truly random form of random access.

The price paid in achievable throughput due to simplicity does offer superior delay performance for lightly-loaded conditions, that is, when S is small. The expected packet delay is comprised of three factors: delay due to transmission τ, delay due to collisions, and delay due to propagation T_R. Though transmission time and propagation delay are fixed, delays due to collisions require special attention. If a user suffers a collision and immediately retransmits, it is highly likely that the user will experience another collision. Thus we mandate that if a user suffers a collision, it must wait a random uniformly distributed delay $\mathcal{U}([0, K])$ before retransmitting. With the analysis formulated in [3] we find the expected number of required retries r for $K \gg 1$ to be

$$\mathrm{E}\{r\} \approx \exp(2G) - 1 \tag{1.10}$$

with expected delay T_c per collision to be

$$\mathrm{E}\{T_c\} = T_R + \frac{(K+1)\tau}{2}. \tag{1.11}$$

Assembling the three components of delay we find the expected delay for the successful transmission of a newly generated packet to be

$$\begin{aligned} \mathcal{T}_{\text{ALOHA}} &= \tau + \mathrm{E}\{r\}\,\mathrm{E}\{T_c\} + T_R \\ &\approx \tau + [\exp(2G) - 1]\left[T_R + \frac{(K+1)\tau}{2}\right] + T_R. \end{aligned} \tag{1.12}$$

1.4.2 Slotted ALOHA

In the last section we learned the poor performance of pure ALOHA is directly related to the high likelihood of a user's packet transmission colliding with those of other users. We also learned that this probability of collision is directly proportional to the vulnerable period of each packet where, for pure ALOHA, the vulnerable period is 2 [packet lengths]. If we are able to decrease this vulnerable period, maximum achievable throughput must increase.

As one would assume from the name, *Slotted ALOHA* (S-ALOHA) further develops the ALOHA technique by mandating the system synchronize packet epochs. The channel is segmented into time units of τ [sec] called slots. A packet arriving into a user's buffer must wait until the beginning of the next slot before attempting ALOHA transmission. Although it may seem this extra delay will decrease the performance of S-ALOHA, the restriction of packet transmission epochs ensures that if a collision occurs, the colliding packets completely overlap each other. Because we no longer must worry about the occurrence of partial overlaps, our only concern is the probability of a packet transmission request during the vulnerable period of τ. Since S-ALOHA has vulnerable period that is half of pure ALOHA, we must revisit (1.7) by noting

$$\begin{aligned} \Pr[\text{success}] &= \Pr[k = 0, t = 1] \\ &= \exp(-G). \end{aligned} \tag{1.13}$$

Following on we see overall throughput is now

$$S = G \exp(-G) \tag{1.14}$$

with maximum achievable throughput

$$S_{\max} = \frac{1}{e} \approx 0.368. \tag{1.15}$$

Thus for the 100 [kbits/sec] channel example from before, with S-ALOHA we will be able to convey approximately 36.8 [kbits/sec] of useful information.

The expected transmission time experienced per newly generated packet for S-ALOHA follows directly from the development for pure ALOHA. Following (1.13), the new expected number of retries for S-ALOHA is

$$\mathrm{E}\{r\} \approx \exp(G) - 1. \tag{1.16}$$

The additional expected delay of waiting for the next available slot must be taken into consideration with expected delay T_c per collision

$$\mathrm{E}\{T_c\} = \frac{\tau}{2} + \frac{K\tau}{2} + \tau + T_R. \tag{1.17}$$

The total expected delay for a newly generated packet is thus

$$\begin{aligned} \mathcal{T}_{\text{S-ALOHA}}(\lambda, l_p, C) &= \frac{\tau}{2} + \tau + \mathrm{E}\{r\}\,\mathrm{E}\{T_c\} + T_R \\ &\approx \frac{3\tau}{2} + [\exp(G) - 1]\left[T_R + \frac{(K+2)\tau}{2}\right] + T_R. \end{aligned} \tag{1.18}$$

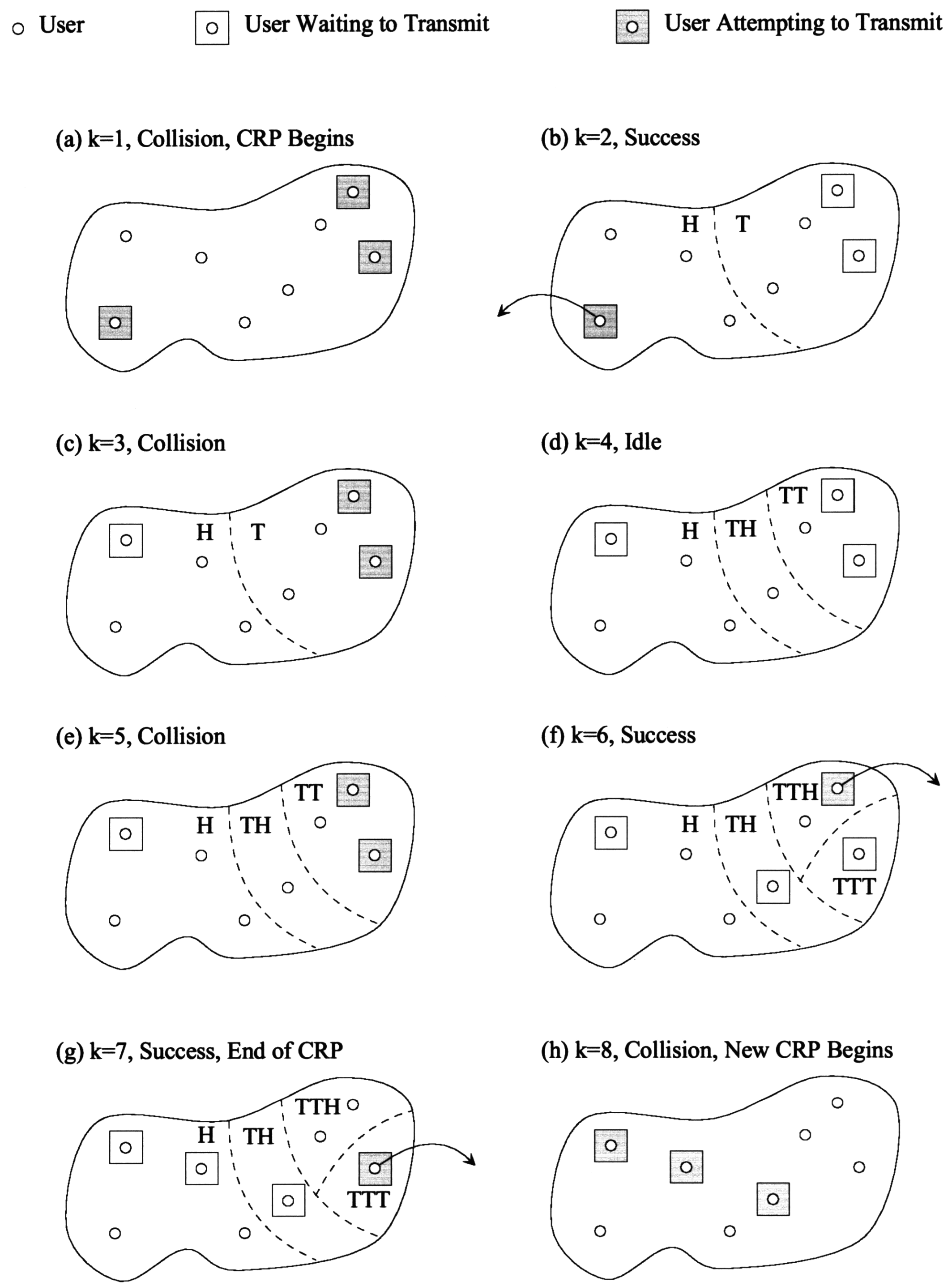
1.4.3 Splitting Algorithms

When a collision is experienced with an ALOHA channel, the users involved in the collision randomly wait and try again. One can imagine that as collisions are occurring, additional new packets are being generated and offered to the channel. The combination of new packets and retransmitted packets saturate the channel and force the probability of packet success given by (1.13) towards zero. Thus the ALOHA channel is susceptible to instability when operated near maximum achievable throughput.

The instability of ALOHA stems from the fact that the algorithm makes no effort to control the admission of new arrivals to the channel. Additionally, the ALOHA algorithm does not try to coordinate the retransmissions of colliding users, resolution is only a probabilistic outcome of the random retransmission delays. *Splitting algorithms*, on the other hand, make an effort to regulate the channel traffic by enforcing a *collision resolution algorithm* which not only sorts out collisions between users, but also blocks new packets from entering the channel until outstanding collisions are resolved.

The first splitting algorithm was the *tree algorithm*[3][4][5][6]. As with the ALOHA protocol, a collision occurs whenever two or more users attempt to transmit their packets in the same slot. The tree algorithm forces all users to enter a *collision resolution period* (CRP) and requires the set of colliding users to split into two separate sets by random coin flipping. The set of users who come up "heads" (H) retransmit while the set of users who came up "tails" (T) wait.

Consider the example CRP depicted in Fig. 1.3. In (a) we show the state of the user population at time $k = 1$ where we see three users have attempted to simultaneously transmit in the same slot resulting in a collision. With all users sensing the collision, the colliding users randomly split into two sets as in (b). Since only one user belongs to the H set the transmission at $k = 2$ is successful and at time $k = 3$ the users in the T set collide once again, shown in (c). In (d) the T set splits into two subsets TH and TT with the set TH being empty. Thus at time $k = 4$ the channel is idle. At time $k = 5$ the TT set accesses the channel resulting in



(a) k=1, Collision, CRP Begins

(b) k=2, Success

(c) k=3, Collision

(d) k=4, Idle

(e) k=5, Collision

(f) k=6, Success

(g) k=7, Success, End of CRP

(h) k=8, Collision, New CRP Begins

Figure 1.3: An abstract representation of a collection of users is shown attempting to access a common channel. An example collision resolution period (CRP) begins in (a) at time $k = 1$. The evolution of the splitting algorithm is shown in (b) through (g) for time $k = 2$ through $k = 7$. Note the new arrivals occurring at times $k = 3$, $k = 6$, and $k = 7$ which result in the initiation of a new CRP in (h).

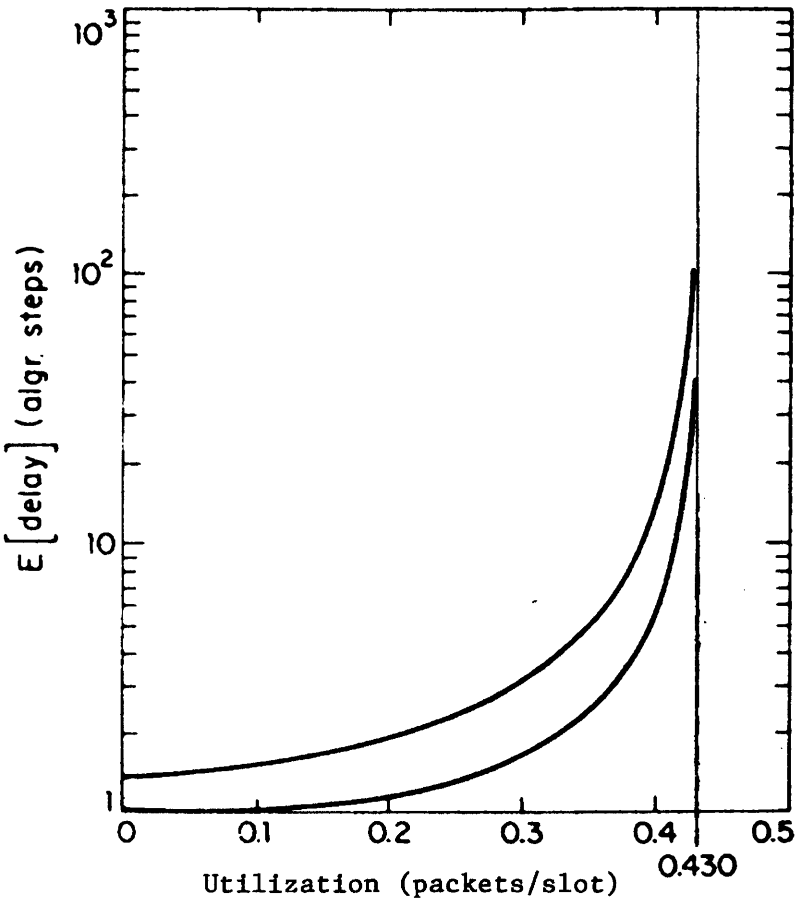


Figure 1.4: Expected delay versus throughput for the tree algorithm. Delay is expressed in algorithm steps which correspond to a single slot duration assuming immediate feedback and no propagation delay.[5]

an additional collision shown in (e). Finally in (f) and (g) this contention is resolved once the nonempty sets TTH and TTT are formed and the CRP is complete. Note however, the new arrivals occurring during the CRP contend for the channel at time $k = 8$ and, as shown in (h), a new CRP is initiated.

With this collision resolution discipline, the tree algorithm is able to obtain a maximum channel throughput of 0.43. Expected delay versus throughput is difficult to analyze for the tree algorithm. Complex upper and lower bounds have been developed in [5] and are shown in Fig. 1.4 under the assumption of Poisson arrivals from an infinite population. Here delay is expressed in algorithm steps which correspond to a single slot duration assuming immediate feedback and no propagation delay. A practical implementation of the tree algorithm would constrain each algorithm step to be at least $T_R + \tau$ [sec] for a packet of length τ [sec].

Variations on the basic tree algorithm exist and are discussed in [7]. The most notable improvement is the *first-come first-serve* (FCFS) splitting algorithm. By using the packet time of arrival (as opposed to random coin flipping) to determine the subsets of a CRP, the FCFS splitting algorithm is shown to have a stable throughput of 0.4871. Enhancements to the FCFS algorithm have shown stable throughput of 0.4878. Finally, yet another modification has improved maximum stable throughput to 0.48780036 where the study of the algorithm departs from engineering and enters the domain of mathematics.

1.4.4 Carrier Sensing

With the above formulation of random access, maximum achievable utilization is generally upper-bounded by the performance of the tree algorithm and its variants. To move beyond this frontier requires rethinking the initial principles of random access thus conveyed. For instance, with the development of pure ALOHA we assume a user with a packet arrival may transmit at will. After all, in (most) human conversations, people wait until the other has finished talking before speaking their minds. Isn't it possible to enforce the same discipline for ALOHA, that is, the user with an arrival first listens for channel activity and only once the channel is silent, sends its packet? This is, in fact, a very viable solution and is the basis of *carrier-sensing multiple access* (CSMA).

With CSMA, a user requesting to send first checks for current channel activity (i.e. a modulated carrier) and only sends when no carrier is detected. Collisions are not entirely avoided, however, since when the channel initially becomes available there may be more than one user seizing the opportunity to send data. The performance of CSMA is highly dependent on the propagation delay associated with the communication resource. Imagine two users both wishing to access a channel with a propagation delay of T. User A experiences a message arrival, waits for the channel to become free (i.e. no detectable carrier), and transmits its message. There is an *uncertainty period* of T during which User B will not detect User A's transmission. If during this period User B experiences a message arrival, it will ascertain the channel to be available and consequently begin transmitting. A collision occurs and both users must resend until collision is avoided.

Because of the uncertainty period described above, CSMA finds application on channels that have small delay T. The analysis of CSMA is similar to that of the ALOHA techniques, that is, we define S as the effective throughput of the channel (successful transmissions) and G as the offered load (new arrivals and necessary retransmissions). Additionally, we define the constant

$$a = \frac{T}{\tau} \tag{1.19}$$

where τ is the packet duration in [sec]. The factor a is a measure of the *uncertainty period* of the CSMA channel. The relation between offered load G and throughput S is developed in [8] and shown to be

$$S = \frac{G \exp(-aG)}{G(1+2a) + \exp(-aG)} \tag{1.20}$$

Thus we see the dependence of the achievable throughput S on a which implies, for a given propagation delay T, we can select τ to meet required throughput requirements. In Fig. 1.5 we show the achievable throughput given by (1.20) for values of $a \leq 1$.

For channels with small T, such as an Ethernet connecting multiple computers together in a local area network, $a \ll 1$ and, from Fig. 1.5, we see CSMA can achieve upwards of 0.9 maximum utilization. The situation is quite different for satellite channels, however. With T large relative to packet length τ, $a \gg 1$, and Fig. 1.5 shows achievable capacity utilization to be much less than 0.18. Hence CSMA is not applicable to satellite communication systems as schemes such as S-ALOHA and the tree algorithm offer better performance.

1.5 Demand Assignment Multiple Access Protocols

Random access works well for efficiently transmitting messages which arrive sporadically. We've seen random access techniques offer small delay when messages are short and system loading is low. Difficulty arises, however, when users wish to sporadically send large messages. In this case we may either increase the slot length to accommodate the larger message size or allow users to split messages and transmit in multiple consecutive slots. Unfortunately both of these fixes lead to poor system performance. The first idea will create inefficiencies for the transmission of shorter messages (due to filler bits necessary to fill the slot size) while the second idea will

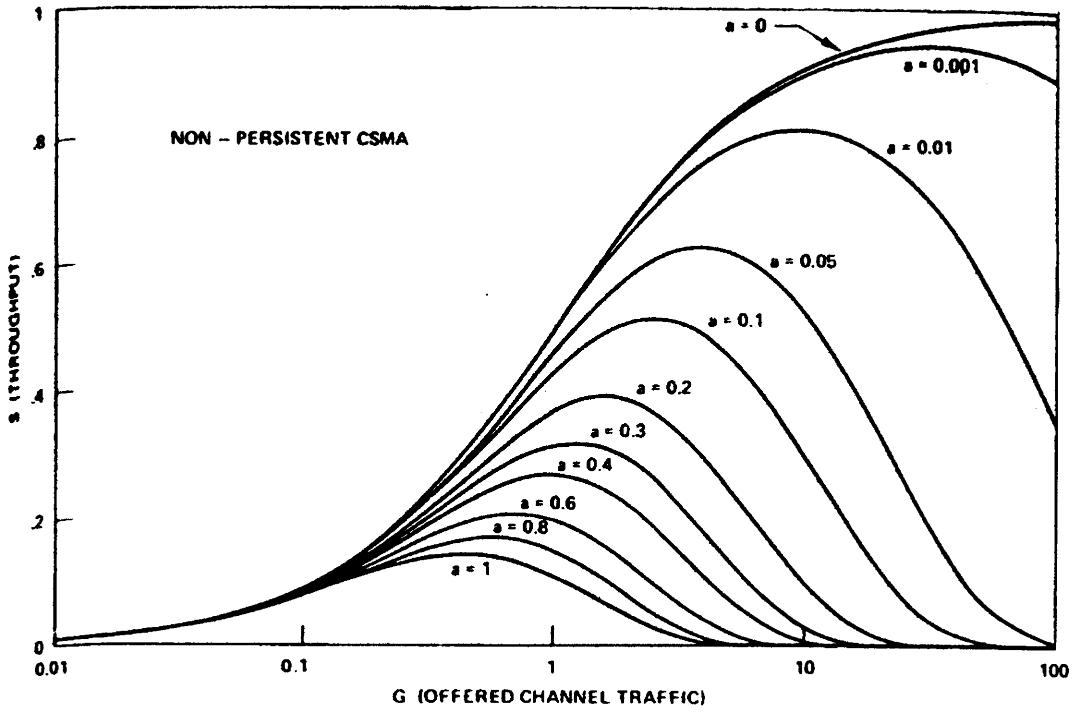


Figure 1.5: Throughput S [packets/packet slot] versus offered traffic G [packets/packet slot] for nonpersistent CSMA for various values of a. Note the offered traffic G represents the arrival rate of new packets and rescheduled packets, but does not imply the actual transmission attempt rate. Notice how dramatically throughput decreases as a becomes large.[8]

increase the message vulnerability time and consequently increase the likelihood of collisions with messages of other users.

A more effective solution to the problem of transmitting longer messages is the idea of assigning contention-free communication resources upon user demand, i.e. *demand assignment.* The idea is to create a division of communication resources to handle two processes. The first process handles requests for resources and is called the *orderwire channel* or *reservation channel.* The second process handles the actual transmission of the user's message and is commonly referred to as the *payload channel* or *data channel.* A *scheduler* is responsible for receiving requests for service from users on the orderwire, acknowledging their requests, coordinating the usage of the data channel resources according to some service discipline, and notifying users when they have access to data channel resources for message transmission.

The goal of the scheduler is to assign the system's communication resources to users in the most efficient manner, that is, the scheduler strives to *statistically multiplex* the traffic requirements of all users onto the available communication resources. Historically the scheduler has been physically associated with the satellite system's ground-based supporting infrastructure. This is because most satellites implementing demand assignment are transponding satellites. With the move to on-board processing satellites, we are now able to place the scheduler on-orbit within the communication payload.

1.5.1 Demand Assignment for Sessions

Demand assignment has been in service with various transponding satellite communication systems for many years. The general scenario is a system that allows a user to establish a low data rate channel for exclusive use and, upon completion of usage, return the channel to the available pool of communication resources. The user communicates to the scheduler (generally ground-based) via the orderwire to set-up and tear-down the connection. Once the circuit is established the user may utilize the channel as desired, but since the channel assignment is exclusive, no other user may access any unused channel capacity. Hence the overall utilization of a circuit-switched demand assignment system is dependent upon the utilization of the individual users of the system.

Circuit-switched demand assignment, as alluded to above, presents serious system inefficiencies if users do not fully utilize their channel assignment. For services such as digitized voice or large file transfer, the channel assigned to the user is typically fully loaded, that is, all available channel capacity is being used. If the service utilizing the channel does not offer a constant *stream* of information, whatever remaining unused channel capacity is lost. This latter situation, namely services which are bursty in nature, is what limits overall system performance as today's user requirements move towards client/server data communications.

1.5.2 Demand Assignment for Messages

We learned in the beginning of this chapter that with multiple access communications, in general, if we are interested in improving utilization, we must be willing to incur some additional delay. The same is true for demand access. The limit of system utilization associated with establishing communication resources for an entire session can be overcome by establishing resources just for sending a single message or a group of messages. In this way, a user will only be occupying the communication resource while it has useful data to send and will be relinquishing the resource to other users when not immediately needed. This rapid exchange of resources yields high overall system utilization. Of course the price paid in this situation is that of delay. Each user must now obtain communication resources on a per message (or group of messages) basis, not just once for the entire session as described above. In order to help mitigate the overhead associated with per message demand assignment, it is highly advantageous to place the scheduler on-orbit thus saving T_R [sec] of delay for each scheduler request.

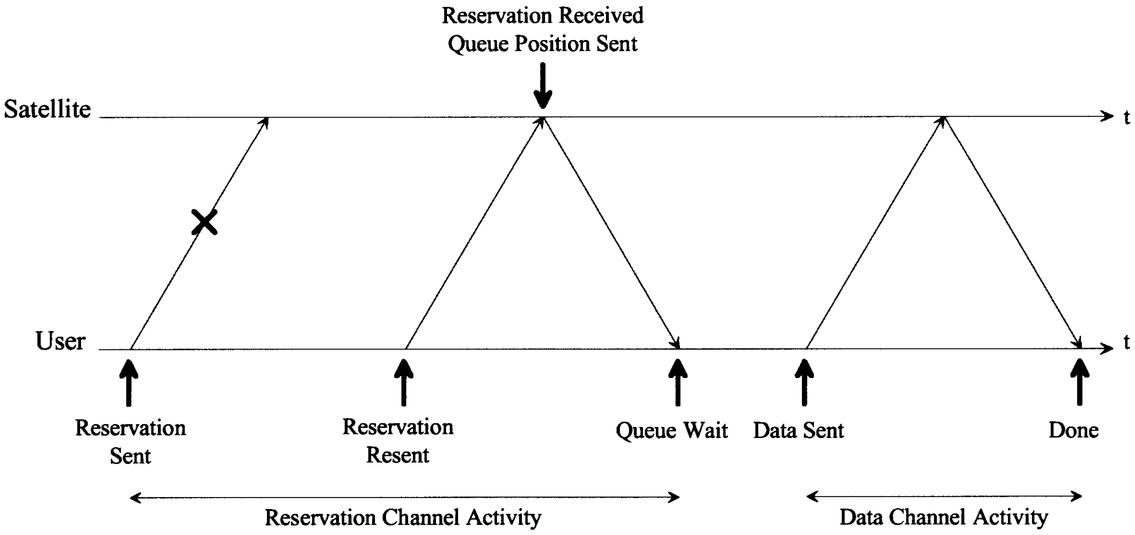


Figure 1.6: Example PRMA exchange between user and satellite. Note since we are using random access on the reservation channel, the user's first attempt to secure a reservation suffered a collision. The message transmission, however, is contention-free and no collisions will occur.

1.5.3 Packet Reservation Multiple Access

Overview

In conversation, the term "demand assignment" generally refers to a system which offers resources for sessions. On the other hand, the term "reservation" refers to resource allocation for messages. *Packet reservation multiple access* (PRMA) [9] is a demand assignment scheme in which a user places a reservation for communication resources to send some message, a message that may be one or more packets in length.

PRMA is a higher level construct that specifies the access discipline for two channels, namely the reservation channel and the data channel. We may analyze any desired combination of multiple access techniques presented above or elsewhere. We also note that we are able to mix orthogonal and non-orthogonal multiple access schemes together in the same system. The discipline to enforce on the data channel is trivial–perfect scheduling on this channel is achieved with statistical time-division multiplexing organized by the scheduler. The reservation channel, on the other hand, may be operated with either an orthogonal or a non-orthogonal (random) multiple access discipline.

Thus we see PRMA allows us to handle bursty message sources while overcoming two major difficulties of random access–large packets required for message transfer and multiple packets for sending large messages. Recall that as the length of a packet increases, the likelihood of collision with other packets increases proportionally. We avoid having to send large packets because we are only implementing random access on the reservation channel. Reservation packets are very small relative to message packets, including only station identification, service request, and possibly traffic priority. Also, only one reservation packet is necessary to request service for a multi-packet message. PRMA is the technique which allows us to achieve statistical multiplexing and random access concurrently. Fig. 1.6 shows an example PRMA exchange between a user and the satellite. Here we are implementing a random access scheme on the reservation channel. Once the short reservation packet is received, the scheduler responds to the request instructing the user when to transmit its message on the data channel. The user waits for its queue allocation and transmits the large message packets on the data channel.

The drawback of PRMA, however, is that each message must incur *two* round-trip delays (i.e. $2T_R$ of extra delay) since each user must send a reservation packet and then send the message

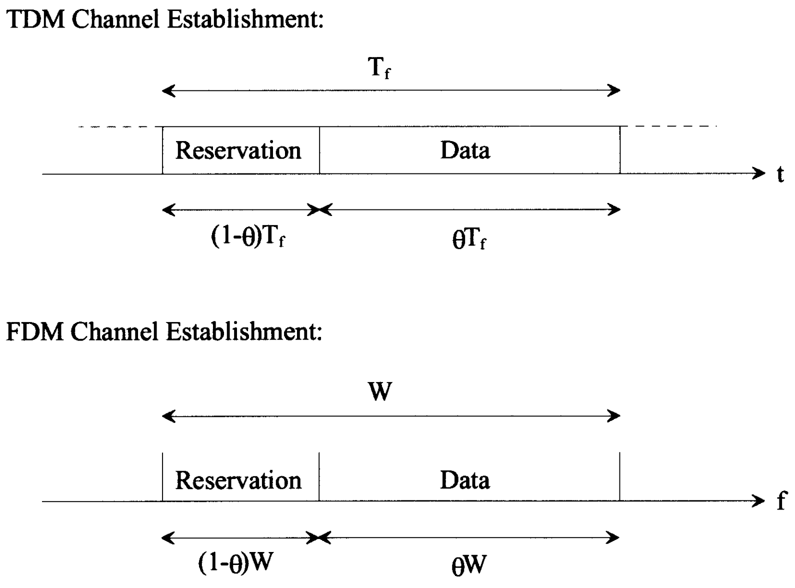


Figure 1.7: We may establish separate channels for reservations and data with either TDM or FDM.

packets. For a lightly-loaded PRMA system, this extra round-trip delay T_R is significant since there is little backlog to the data channel queue. As loading increases, however, the extra delay T_R experienced is less of an issue since the data packets must wait for service in the data channel queue.

Establishing Two Channels

As with all multiplexing, we may separate the reservation channel and data channel either in time or frequency, as shown in Fig. 1.7. If our only method of multiplexing is TDM, we must assign some proportion θ of available time to the data channel and consequently the remaining $1 - \theta$ of available time to the reservation channel. In this way we have created a reservation interval and data transfer interval from some larger TDM frame.

With the availability of FDM, we are able to assign a proportion θ of available frequency to the data channel and the remaining $1 - \theta$ of available frequency to form the reservation channel[10]. For a channelized system we simply assign a fixed number of available channels to data and the remaining channels to reservations.

Channel Access Disciplines

Once the two separate channels are established for the reservation process and data transfer process, we must next determine the multiple access scheme to be implemented on each. We are free to choose whichever access discipline is appropriate for the two channels independently. As alluded to above, with the interest of handling the transfer of large messages with sporadic arrivals, we generally want to incorporate random access on the reservation channel for its low delay property while using perfectly-scheduled TDM on the data channel for its optimal throughput and delay characteristics.

Other combinations of multiple access techniques are possible as well. For instance, we may wish to implement TDMA on the reservation channel such that there is no channel contention while making reservations. In this case, each user is assigned a specific slot within the TDMA frame on the reservation channel. As imagined, when a user is interested in accessing the data channel, it must first wait for its reservation time slot before requesting service. We will see in

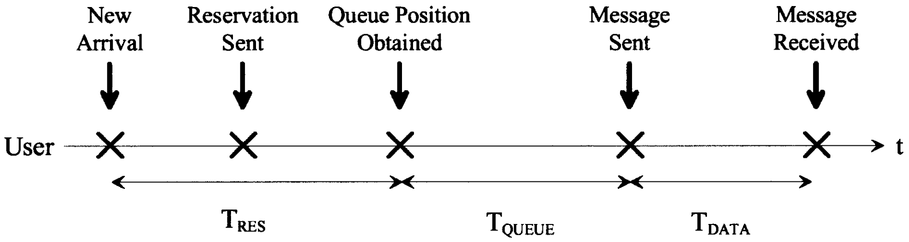


Figure 1.8: The delay associated with message transfer using PRMA is the sum of delay associated with placing the reservation and delay associated with the data channel queue and message transmission.

the next chapter that this delay becomes significant as the number of supported users increases.

Message Transfer Delay Analysis

To analyze the delay associated with PRMA message transfer, consider the sequence of events shown in Fig. 1.8. The total message delay is the sum of the delay in placing the reservation, the time spent waiting for access to the data channel, and the actual message transmission time, that is,

$$\mathcal{T}_{\mathrm{PRMA}} = \mathcal{T}_{\mathrm{RES}} + \mathcal{T}_{\mathrm{QUEUE}} + \mathcal{T}_{\mathrm{DATA}}. \tag{1.21}$$

The first and third components, namely $\mathcal{T}_{\mathrm{RES}}$ and $\mathcal{T}_{\mathrm{DATA}}$, depend on the access discipline associated with the reservation channel and the data channel, respectively. The delay associated with queuing, $\mathcal{T}_{\mathrm{QUEUE}}$, is a function of the message lengths to be sent on the data channel.

Channel Capacity Assignment

For the efficient operation of PRMA we must intelligently divide available communication capacity between the reservation channel and the data channel. Thus it is necessary to choose θ so that $\mathcal{T}_{\mathrm{PRMA}}$ is minimized for the given channel access disciplines and user traffic statistics. If too little capacity is assigned to the data channel, the service time associated with each message transmission will increase and users will spend more time waiting for the data channel queue to advance. On the other hand, if too little capacity is assigned to the reservation channel, the delay associated with placing reservations will be excessive and will limit the utilization of the data channel queue under light system loading. Additionally, the fact that the user traffic statistics may be time-varying implies the optimal value of θ may be time-varying as well. This problem of choosing θ for efficient operation of PRMA is treated in Appendix B.

1.6 Summary and Introduction to the Following Chapters

The above account describes the evolution of satellite communications and the need for efficient transmission techniques to allow the expansion of both present and future data communication requirements. We have introduced the concept of multiple access communication and briefly described the underlying multiplexing techniques of which multiple access builds upon. Two categories of multiple access techniques are defined, namely orthogonal multiple access and random access. Orthogonal multiple access has the property that a given user's transmissions on the communication channel are isolated from all other users. With random access, on the other hand, users must contend for access to the communication channel and, consequently, collisions between users will occur.

The concept of random access allows for the efficient transmission of messages from bursty sources, typical of data communications. A brief survey of the random access techniques applicable to satellite communications is offered introducing ALOHA transmission techniques and splitting algorithms that allow for the transmission of sporadic messages with low delay. We briefly visited carrier sensing techniques, but found their performance to be poor for the satellite channel with large propagation delay.

With demand assignment multiple access we are able to dynamically assign satellite transmission capacity to users as needed, either for the duration of an entire communication session or for each individual user message that must be transmitted. Since orthogonal multiple access offers high capacity utilization and random access offers low delay, we search for a method of efficiently combining the two transmission techniques. Demand assignment allows us to reap the benefits of both techniques. We can use random access for the short service request (reservation) packets to minimize delay while using an orthogonal multiple access technique to handle the long message transfer thereby maximizing capacity utilization and throughput.

We use the topics introduced in this chapter to analyze the performance of two candidate uplink architectures in Chapter 2. Here we are interested in determining the delay versus throughput characteristics for the various multiple access schemes discussed above. In addition to the two architectures under investigation, we consider two distinctly different message transmission requirements and assess their performance with both uplink configurations.

In Chapter 3 we discuss packet arrival models in more generality than the assumed Poisson arrival model used in this chapter. Specifically we consider the Markov-modulated Poisson process which has many applications to source modeling and aggregate traffic modeling. On this last note we also briefly discuss recent results on measurements of the statistics of packet arrivals on a local area network and discuss the concept of self-similar traffic.

Finally, we generalize multiple access communications to a vector space problem in Chapter 4. The concept of code division multiplexing is further refined and the equivalency of the three multiplexing techniques introduced above is demonstrated. Slotted ALOHA is also generalized beyond the above description and notions of combining CDMA with Slotted ALOHA are discussed.

Chapter 2

Satellite Uplink Architecture Investigation

2.1 Introduction

System architecture design and evaluation is key to developing communication systems that effectively meet the needs of both present-day requirements and those of the vision for the future. It is easy to allow designs of previous systems to influence the specification of next-generation communication systems, especially if the designer does not consider the changing communication needs and services of the new era. This issue holds for satellite communication systems as the past has been characterized by large circuit-switched gateways utilizing the satellite for trunking. The future of satellite communications promises the satellite will not only be used for long-haul transmission, but also for network *access* by individual users. In addition, the nature of services provided by satellite is moving away from simple voice services towards that of bursty packet-switched data communications. Thus, in light of these new user requirements, it is exceedingly important to reconsider the simple expansion of past architectures to handle tomorrow's needs.

In this chapter we wish to investigate the performance of two candidate architectures for satellite transmission of messages from bursty sources. The first architecture, which we will call the *channelized uplink* architecture, is based on the idea of providing multiplexing by dividing total transmission capacity into a fixed number of frequency channels. The second architecture to be investigated, the *dynamically-assigned uplink* architecture, strives to offer the largest instantaneous data transmission rate by offering one large-capacity channel dynamically time-shared between users.

The performance of both architectures is considered in the light of two distinctly different user requirements. For the first requirement, delay vs. throughput performance of the channelized and dynamically-assigned architectures is developed for the transmission of large messages of fixed length arriving from bursty sources. This scenario is typical of a user needing to send an electronic mail message or a sensor needing to convey accumulated telemetry. Demand assignment multiple access is considered for both architectures considering fixed-access and random-access reservation channels. We present quantitative results by making real-world parameter assumptions, allowing the development of intuition into the performance of both architectures.

The second user requirement involves the interaction between a client application and its associated server. Most client-server interaction is asymmetric, that is, there is a large data flow from server to client while client to server data flow is small in comparison. The interaction of a world wide web browser client and associated server, for example, is largely asymmetric. A user's transmission requirements are generally short information requests or information acknowledgments. The transmission from the web server to the user's browser, however, generally contains large images and other multimedia information orders of magnitude larger than user request packets. In addition to the asymmetry, the transmission requirements are again bursty in nature, typically initiated by user interaction with the client application.

A model of a client-server session with the above characteristics is developed in this chapter. Session arrival and length statistics are presented as well as packet arrivals within the session. Since our focus is on the satellite uplink, we develop the perfectly-scheduled lower bound on the delay associated with the client to server data exchange for both the channelized and dynamically-assigned uplink architectures. As with the large message transfer analysis, typical parameters are used to offer quantitative results to aide in architecture specification.

2.2 Preliminaries

2.2.1 Terminology

Before beginning our analysis it would be advantageous to clearly define the vocabulary used throughout this chapter.

User Analogous to *station* or *node*, the entity that both sends and receives messages through the satellite communication resource.

Terminal The device a user operates to access an on-orbit satellite communication resource.

Session A *session* is considered to be the time span from when a user initiates usage with some resource to when the user finishes all interaction with the resource.

Message The most basic form of the information generated by a user needing to be conveyed is termed the *message.*

Packet Small messages are typically referred to as *packets.* Additionally, a large message may be divided into multiple packets before transmission.

Scheduler As described in Chapter 1, the *scheduler* coordinates the usage of a communication channel or channels between multiple users.

2.2.2 Architecture Descriptions

Channelized Uplink Architecture

One method for offering communication resources to multiple users is to divide the transmission medium into multiple equal-capacity channels. These channels are then assigned to users on a need basis, viz., we implement demand assignment as discussed in Chapter 1. When these channels are created by a fixed division of available transmission frequency spectrum we have the familiar frequency division multiplexing. The *channelized uplink architecture* is a division of available uplink capacity into N_c individual channels with any user occupying only one channel at a time.

Dynamically-Assigned Uplink Architecture

An alternative technique to sharing a communication resource is statistical multiplexing. Here we avoid the temptation of dividing the available uplink capacity into multiple channels and simply keep the available uplink capacity intact as one large channel. With the *dynamically-assigned uplink architecture* we are dynamically assigning access rights to the uplink capacity to one user at a time. Note the time division is dynamic, we are not implementing TDMA where each user is assigned a static time slot in some fixed frame duration.

2.3 Large Message Transfer Performance

2.3.1 Overview

An important performance evaluation of the two architectures under consideration is to evaluate their effectiveness for transferring large messages. In this section we determine the expected delay

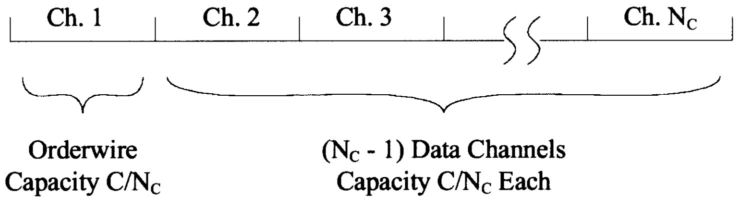


Figure 2.1: DAMA for the channelized uplink assigns one of the N_c uplink channels as the orderwire leaving the remaining $N_c - 1$ channels available for data transfer.

to transfer a message of L [bits] through systems with channelized and dynamically-assigned uplink architectures. The analysis considers delays associated with using various multiple access techniques in addition to offering the theoretically optimal perfect scheduling lower bound to delay. We begin by stating our model and necessary assumptions to simplify analysis. Next we develop the analytical expressions to evaluate the two architectures under investigation with generic parameters and then specialize to demonstrate their performance with typical real-world values. Following the presentation of results we summarize the important characteristics of both architectures.

2.3.2 Message Arrival Model

The source statistics for this message analysis model assume fixed length messages of length L [bits]. These messages originate from an infinite population of users according to a Poisson arrival model of composite arrival rate λ_c. Alternatively, if we were interested in modeling a finite user population, we could use the binomial arrival model. The added effort of adopting the binomial model is generally not required, however, since the results obtained for large user populations converge to those of the Poisson model[11].

2.3.3 Channelized Uplink Architecture

Demand assignment multiple access (DAMA) is investigated with the channelized uplink architecture by assigning one of the N_c channels as an orderwire thus leaving the remaining $N_c - 1$ channels as data channels for message transmission. With this arrangement, as shown in Fig. 2.1, an uplink of total capacity C [bits/sec] is divided into N_c channels each of capacity $\frac{C}{N_c}$ [bits/sec]. As described in Chapter 1, a user wishing to send a message must first place a reservation with the scheduler. The scheduler assigns the available data channels in a first-come first-served (FCFS) discipline. Once a user has completed the message transmission through the data channel, the channel is returned to the available resource pool for use by other users.

As just described, the scheduler achieves system multiplexing by controlling access to the data channels. We must, however, enforce discipline on the scheduler's orderwire as well. We can use any multiple access scheme desired on the orderwire, broadly categorized as being either orthogonal (such as TDMA, FDMA, etc.) or random-access (such as ALOHA, Tree Algorithm, etc.) Since the orthogonal multiple access schemes are static (a user is always assigned a unique slot or channel), we refer to this configuration as 'fixed-access' and retain the terminology of 'random-access' to describe the two orderwire disciplines under consideration.

Fixed-Access Orderwire

One technique of facilitating multiple access on the orderwire channel is to use a fixed-access protocol such as TDMA. In this scenario, each user is assigned a time slot in a reservation frame and hence we define N to be the number of supported users. If we assume each user's reservation packet is of length l_r [bits] and the channel capacity of the orderwire channel is

$C_r = \frac{C}{N_c}$ [bits/sec] then we can compute the expected setup time to be[1]

$$D_{\text{SU,FA}} \; = \; \mathcal{T}_{\text{TDMA}}\left(\frac{\lambda_c}{N}, l_r, l_r, N, \frac{C}{N_c}\right). \tag{2.1}$$

With this configuration, each user is always assured of having a time slot in the reservation frame to request service for a new message arrival. The reservation needs only to be sent once since there is no contention on the orderwire channel.

Random-Access Orderwire

To achieve multiple access on the orderwire in a contention-mode manner we consider S-ALOHA. This random-access orderwire will have expected setup time of

$$D_{\text{SU,RA}} \; = \; \mathcal{T}_{\text{S-ALOHA}}\left(\lambda_c, l_r, \frac{C}{N_c}\right). \tag{2.2}$$

Of course, as with any random-access scheme, it may be necessary for a user to retransmit its reservation packet if channel contention is experienced.

Service Wait Time and Transmission Time

Once the reservation is received, the scheduler determines the earliest availability of a data channel and notifies the user when it may begin sending. The transmission time of the actual message, once the queue is available is simply

$$D_{\text{TX}} \; = \; \frac{LN_c}{C}. \tag{2.3}$$

To determine the expected wait for an available channel we realize the data channels appear as an M/D/k queue with $(N_c - 1)$ servers each operating at $\frac{C}{N_c}$ [bits/sec] to handle message arrivals of rate λ_c.

Overall Message Delay

With all of the above components described, the expected overall message delay for either orderwire discipline is of the form[2]

$$\mathcal{T}_{\text{FDM}} \; = \; D_{\text{SU}} + \mathcal{Q}_{\text{M/D/k}}\left(\lambda_c, \frac{1}{D_{\text{TX}}}, N_c - 1\right) + D_{\text{TX}} + \mathcal{T}_R \tag{2.4}$$

where D_{SU} is taken as either (2.1) or (2.2) depending on the desired orderwire discipline.

2.3.4 Dynamically-Assigned Uplink Architecture

Demand assignment is also considered for the dynamically-assigned architecture. There are two ways of creating a mechanism to receive reservations. First, we can create a frame structure where $\theta\%$ of the frame length is devoted to data transfer and the remainder $(1-\theta)\%$ is devoted to generating a *reservation interval.* The second alternative is to divide the uplink bandwidth into two separate frequency channels as in Fig. 2.2. Such a configuration assigns $\theta\%$ of uplink bandwidth to data transfer and $(1-\theta)\%$ to a *reservation channel.*

In either case there is simply a division of the available uplink capacity C between the two required services. Both techniques are shown to have nearly the same throughput and delay characteristics, however the technique of utilizing a frequency-separated reservation *channel* is more easily analyzed and therefore adopted here[12]. As with the orderwire channel of the channelized uplink architecture investigated above, we have the option of operating the reservation channel in either fixed-access or random-access fashion, which we develop next.

[1] We use the functional description $\mathcal{T}_{\text{TDMA}}(\cdot)$ to denote the expected transmission delay for TDMA and $\mathcal{T}_{\text{S-ALOHA}}(\cdot)$ for the expected transmission delay for S-ALOHA, as developed in Chapter 1.

[2] Note our use of the functional queuing delay models $\mathcal{Q}_{\text{M/D/k}}(\cdot)$ and $\mathcal{Q}_{\text{M/D/1}}(\cdot)$ is described in Appendix A.

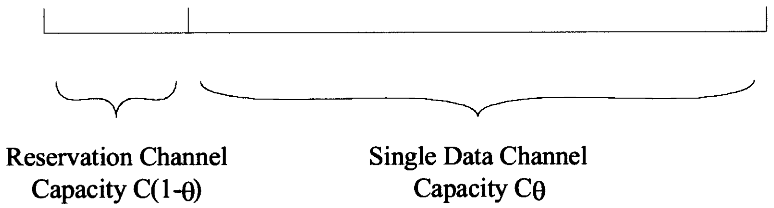


Figure 2.2: DAMA for the dynamically-assigned uplink assigns $1 - \theta$ % of available uplink capacity to reservations while the remaining capacity is devoted to the data channel.

Fixed-Access Reservation Channel

For fixed-access, as above, we consider the use of TDMA on the reservation channel. Reservation packets are again l_r [bits] long and we note the available reservation channel capacity to be $C_r = (1 - \theta)C$. The expected setup time is thus

$$\mathcal{D}_{\text{SU,FA}} = \mathcal{T}_{\text{TDMA}}\left(\frac{\lambda_c}{N}, l_r, l_r, N, (1-\theta)C\right) \tag{2.5}$$

for N accessing users, as we've seen before.

Random-Access Reservation Channel

To achieve random-access reservations we operate the reservation channel in contention mode using S-ALOHA. The expected delay follows from (2.2) with $C_r = (1-\theta)C$. The expected setup time for a user to secure a reservation is therefore

$$\mathcal{D}_{\text{SU,RA}} = \mathcal{T}_{\text{S-ALOHA}}\left(\lambda_c, l_r, (1-\theta)C\right). \tag{2.6}$$

Service Wait Time and Transmission Time

Transmission time of the entire message, considering the absence of capacity assigned to the reservation channel, is given by

$$\mathcal{D}_{\text{TX}} = \frac{L}{\theta C}. \tag{2.7}$$

Immediately, while comparing (2.7) with (2.3), we note the possibility of smaller overall message delay with the dynamically-assigned architecture. The data channel appears as an M/D/1 queue operating at θC [bits/sec] handling message arrivals of Poisson rate λ_c.

Overall Message Delay

Appropriately combining the above components yields the expected overall message delay

$$\mathcal{T}_{\text{TDM}} = \mathcal{D}_{\text{RES}} + \mathcal{Q}_{\text{M/D/1}}\left(\lambda_c, \frac{1}{\mathcal{D}_{\text{TX}}}\right) + \mathcal{D}_{\text{TX}} + \mathcal{T}_R. \tag{2.8}$$

As before, the placeholder $\mathcal{D}_{\text{RES}}$ reflects the expected reservation delay described by either (2.5) or (2.6) for the two reservation channel access disciplines.

2.3.5 Perfect Scheduling, Slotted ALOHA, and TDMA Baseline Cases

In addition to the analysis of the two architectures above, we wish to investigate three transmission techniques that form baselines to assess the performance of the demand assignment architectures under investigation.

Perfect Scheduling

We can determine the absolute minimum bound on expected transmission delay if we assume *perfect scheduling*. Perfect scheduling is the idealization of assuming all users have full information of the state of the system, are able to coordinate with each other to determine scheduling, and are aware of the exact moment the data channel is available for transmission. The important fact is that no other transmission scheme can achieve a smaller expected transmission delay than the perfect scheduling assumption.

The expected transmission delay assuming perfect scheduling is simply the time required for a user to transmit a message from source to destination and is given by

$$T_{\mathrm{PS}} = \mathcal{Q}_{\mathrm{M/D/1}}\left(\lambda_c, \frac{C}{L}\right) + \frac{L}{C} + T_R. \tag{2.9}$$

We see T_{PS} is composed only of the delay associated with the raw physical limits of finite transmission capacity C and propagation delay T_R.

TDMA

The orthogonal multiple access technique of TDMA is also investigated to demonstrate the performance of a fixed-access transmission scheme. Here again we are interested in supporting N users wishing to send messages through the uplink capacity of C [bits/sec]. Each user is assigned a slot of l_p [bits] in length in a frame consisting of N slots. The expected delay for a user to send a message of L [bits] is given by

$$T_{\mathrm{TDMA}} = \mathcal{T}_{\mathrm{TDMA}}\left(\frac{\lambda_c}{N}, L, l_p, N, C\right). \tag{2.10}$$

We note as the number of users N increases the frame length T_f also increases, that is, we hold slot length constant.

Slotted ALOHA

Finally, our third baseline case exemplifies the performance of a random access transmission technique. We investigate S-ALOHA with a slot duration of one message length L [bits]. Utilizing all uplink capacity, the expected delay to convey a message from source to destination using S-ALOHA is

$$T_{\mathrm{S-ALOHA}} = \mathcal{T}_{\mathrm{S-ALOHA}}\left(\lambda_c, L, C\right). \tag{2.11}$$

2.3.6 Model Parameters

At this point it is prudent to summarize the parameters used in our model.

Parameter	Description	Units
T_R	Round-trip propagation delay	[sec]
L	Message length	[bits]
l_p	Packet length	[bits]
l_r	Reservation request packet length	[bits]
C	Total uplink channel capacity	[bits/sec]
T_f	Frame length (TDMA only)	[sec]
N	Total number of users of system	[users]
N_c	Number of channels (Channelized Architecture)	[channels]
λ_c	Composite message arrival rate	[msgs/sec]

2.3.7 Assumed Parameter Values

With the analytical development of our model complete, it is now necessary to produce a few examples of message delay performance for the two architectures under investigation. Because of the large number of parameters we must make qualified assumptions to facilitate system performance evaluation by variation of key parameters. The following list highlights parameters of our model which are assigned fixed values either due to physical constrains or practical implementation constraints.

Propagation Delay T_R The round-trip propagation delay T_R is fixed at 0.25 [sec] for all cases. This chosen value is typical for geostationary satellite communication systems and can be changed when considering medium earth orbit (MEO) or low earth orbit (LEO) systems.

Message Length L The fixed-length messages are assumed to be $L = 10$ [kbits] in length. This message length is appreciable compared to other signaling packets (such as acknowledgment packets of popular protocols) and therefore satisfies our large message assumption.

Reservation Packet Length l_r When analyzing demand assignment utilizing reservation request packets, the length of such packets, l_r, is set to 100 [bits]. This value is reasonable assuming the need to convey source and destination station addresses, message length (for a general system), and possibly message priority.

TDMA Packet Length l_p For analysis of TDMA, we assume the entire message of L [bits] is split into packets of length $l_p = 1000$ [bits].

Channelized Architecture Configuration For illustration purposes, the total capacity of the channelized architecture transponder is divided into $N_c = 10$ equal-capacity channels. One channel is assigned as the orderwire channel with the nine remaining channels assigned as data channels.

Dynamically-Assigned Architecture Configuration The dynamically-assigned architecture is configured such that $\theta = 0.9$ thus assigning the same amount of total transponder capacity to data transmission as in the channelized architecture. Note however that this assignment of θ is *not* optimal and, in many cases, the flexibility of capacity assignment in the dynamically-assigned architecture offers further performance gain over the channelized architecture.

2.3.8 Analysis

We are now prepared to analyze the performance of the channelized and dynamically-assigned uplink architectures for transmission of large messages from bursty sources. With assignment of typical values to the auxiliary parameters of our model we focus on the remaining parameters of message arrival rate and total uplink transmission capacity. Although we assume an infinite user population for the random access schemes, we must specify the number of accessing users for the fixed-axis orderwire configurations.

Six transmission techniques are investigated–there are four demand assignment configurations for the architectures under examination and we also include results for TDMA and S-ALOHA. In addition to these transmission techniques, expected message delay assuming perfect scheduling is presented.

Results are shown in Figs. 2.3-2.11 where we have chosen $C \in \{10, 100, 1000\}$ [kbits/sec] and $N \in \{10, 100, 1000\}$ [users]. The figures offer expected message delay vs. normalized message arrival rate which we term *utilization* ρ. We relate utilization and message arrival rate with

$$\rho = \lambda_c \frac{L}{C} \tag{2.12}$$

for the channelized and dynamically-assigned uplink architectures. The following matrix indexes the results as well as offers quick utilization to message arrival rate conversion.

Capacity C [bits/sec]	Users N	$\lambda_{c,max}$ [msgs/sec]	Figure
10k	10	1	2.3
100k	10	10	2.4
1M	10	100	2.5
10k	100	1	2.6
100k	100	10	2.7
1M	100	100	2.8
10k	1000	1	2.9
100k	1000	10	2.10
1M	1000	100	2.11

In each plot, the seven delay-utilization curves are labeled as follows:

Description	Label
Traditional TDMA (Baseline Case)	(1)
Traditional S-ALOHA (Baseline Case)	(2)
Channelized Architecture, Fixed-access Orderwire Channel	(3)
Channelized Architecture, Random-access Orderwire Channel	(4)
Dynamically-Assigned Architecture, Fixed-access Reservation Channel	(5)
Dynamically-Assigned Architecture, Random-access Reservation Channel	(6)
Perfect Scheduling (Lower Bound)	PS
Round-trip Propagation Delay	T_R

2.3.9 Interpretation of Results

TDMA Transmission

Traditional fixed-access is shown with TDMA in Figs. 2.3-2.11 for comparison purposes. TDMA is seen to have superior capacity utilization over all other schemes considered. For the majority of cases, however, implementing TDMA for the transmission of messages from bursty sources results in expected delay values at least an *order of magnitude greater* than the channelized or dynamically-assigned architectures considered. TDMA transmission is a highly undesirable scheme for handling traffic from bursty sources.

Slotted ALOHA Transmission

Conventional random-accessing using S-ALOHA is shown in Figs. 2.3-2.11 to have superior expected delay performance over all other schemes considered. This performance, like in the TDMA case, comes at the price of some other factor, namely the maximum achievable utilization. Recall S-ALOHA has maximum utilization of $\frac{1}{e}$. Thus it appears S-ALOHA and other similar random-access schemes are well suited for handling traffic from bursty sources.

We immediately see higher utilization is achievable with the use of a fixed-access multiple access scheme, whereas the task of minimizing expected delay (for light loading) is facilitated through the use of a random-access multiple access scheme. Multiple access schemes attempting to simultaneously improve utilization and delay performance must somehow combine the techniques of both fixed-access and random-access transmission.

Channelized Architecture, Fixed-Access Orderwire

The performance of the channelized architecture implementing a fixed-access discipline on the orderwire is essentially constant over all utilization values up to about 70% utilization. At this point the expected delay asymptotically approaches 90% utilization, consistent with intuition since 90% of available capacity is dedicated to data transmission.

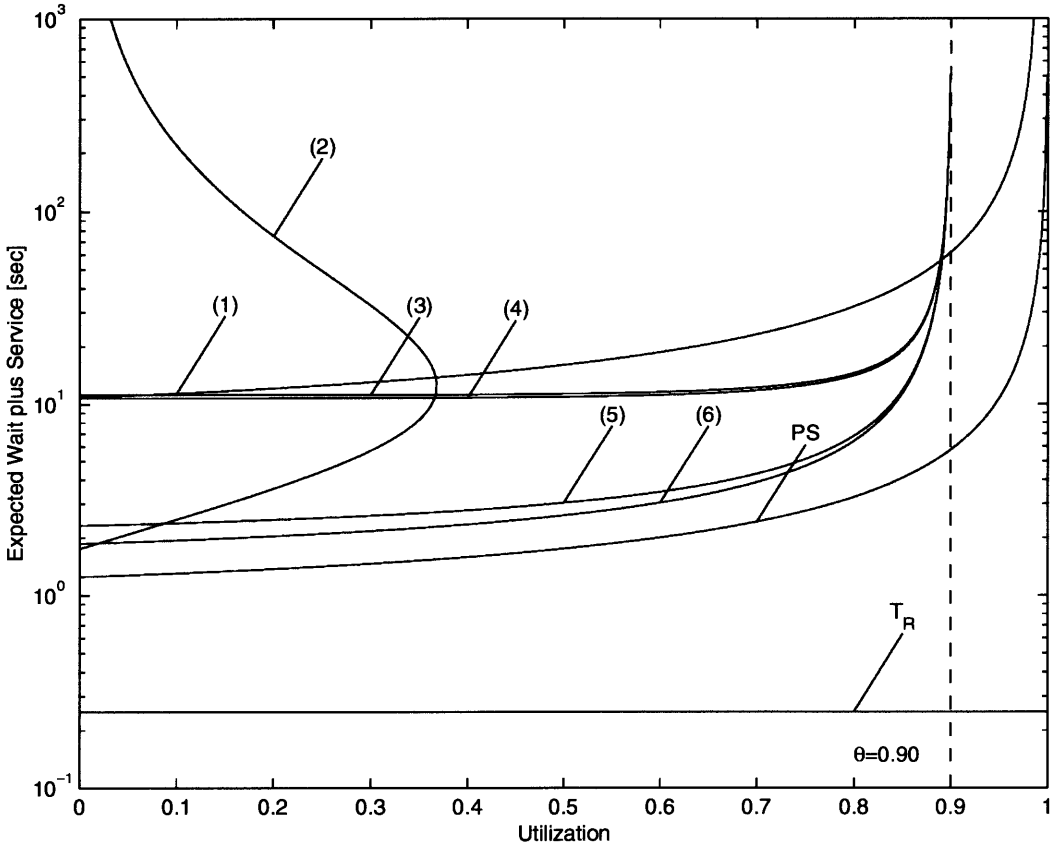


Figure 2.3: $L = 10$ [kb] message transfer for $N = 10$ users at a capacity of $C = 10$ [kbits/sec] with $0 \leq \lambda_c \leq 1$ [msgs/sec]. Note we have assigned 10% of communication resources to placing reservations ($\theta = 0.9$).

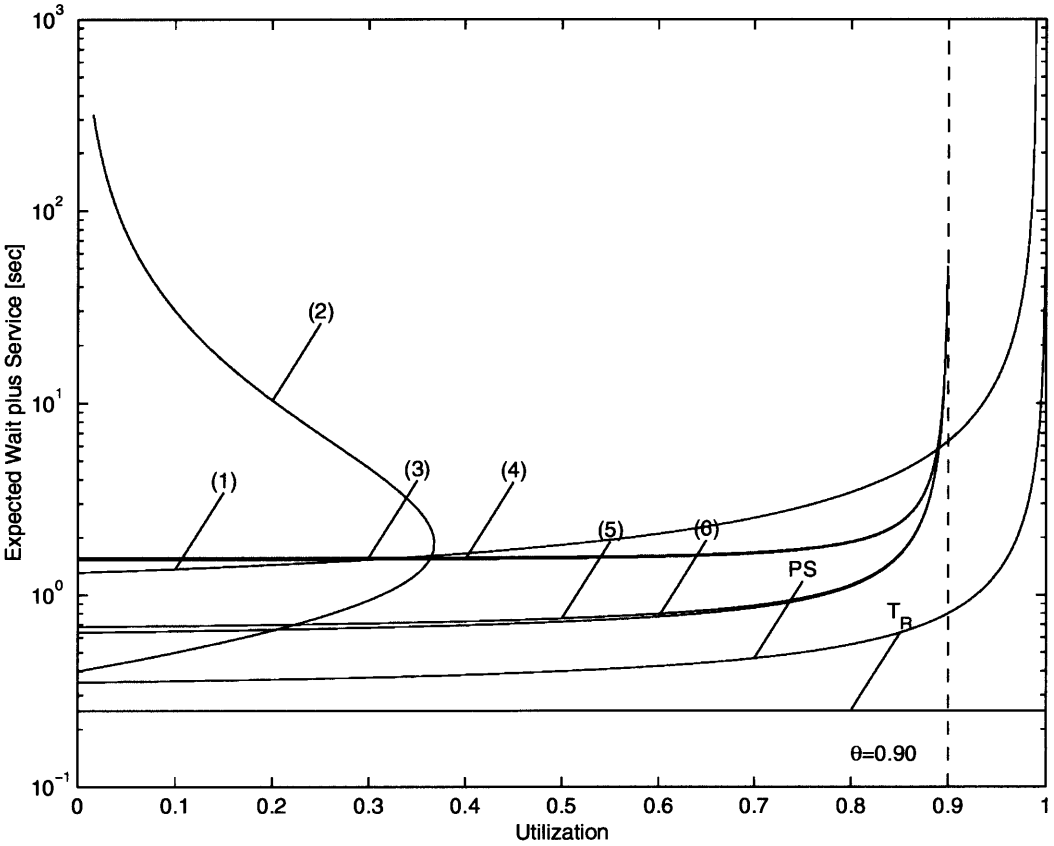


Figure 2.4: $L = 10$ [kb] message transfer for $N = 10$ users at a capacity of $C = 100$ [kbits/sec] with $0 \leq \lambda_c \leq 10$ [msgs/sec]. Note we have assigned 10% of communication resources to placing reservations ($\theta = 0.9$).

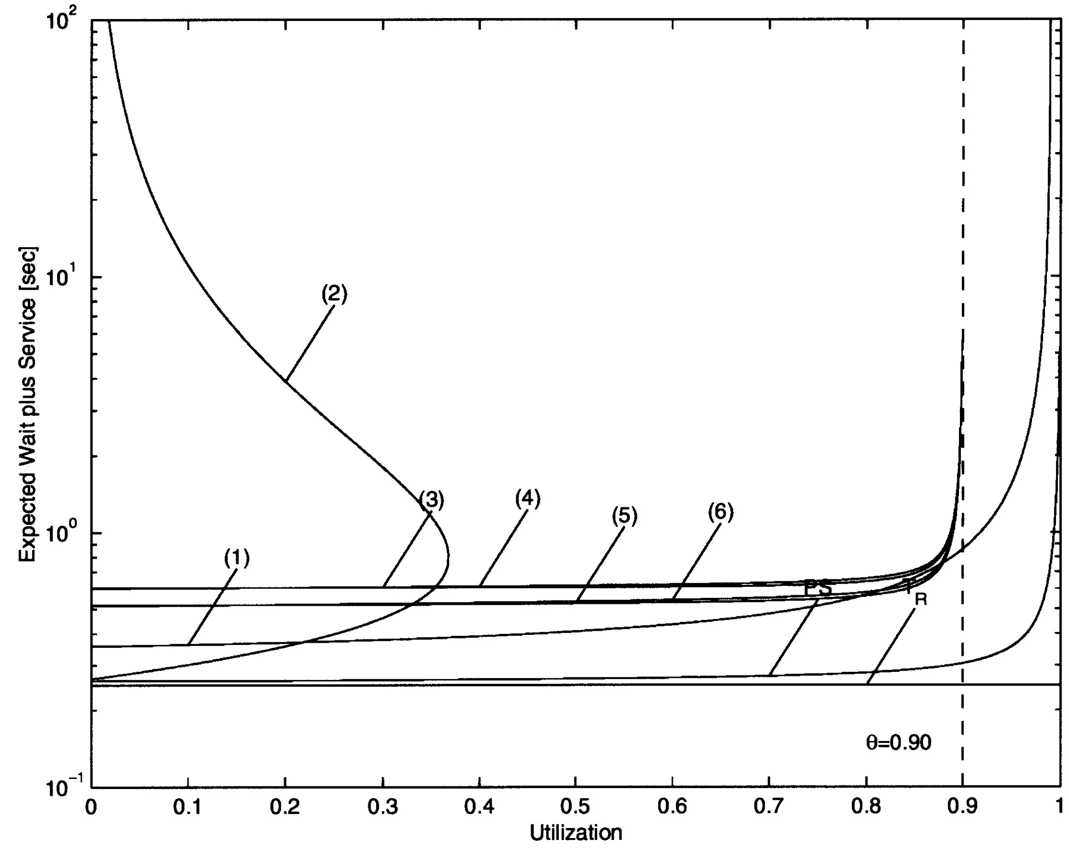


Figure 2.5: $L = 10$ [kb] message transfer for $N = 10$ users at a capacity of $C = 1$ [Mbits/sec] with $0 \leq \lambda_c \leq 100$ [msgs/sec]. Note we have assigned 10% of communication resources to placing reservations ($\theta = 0.9$).

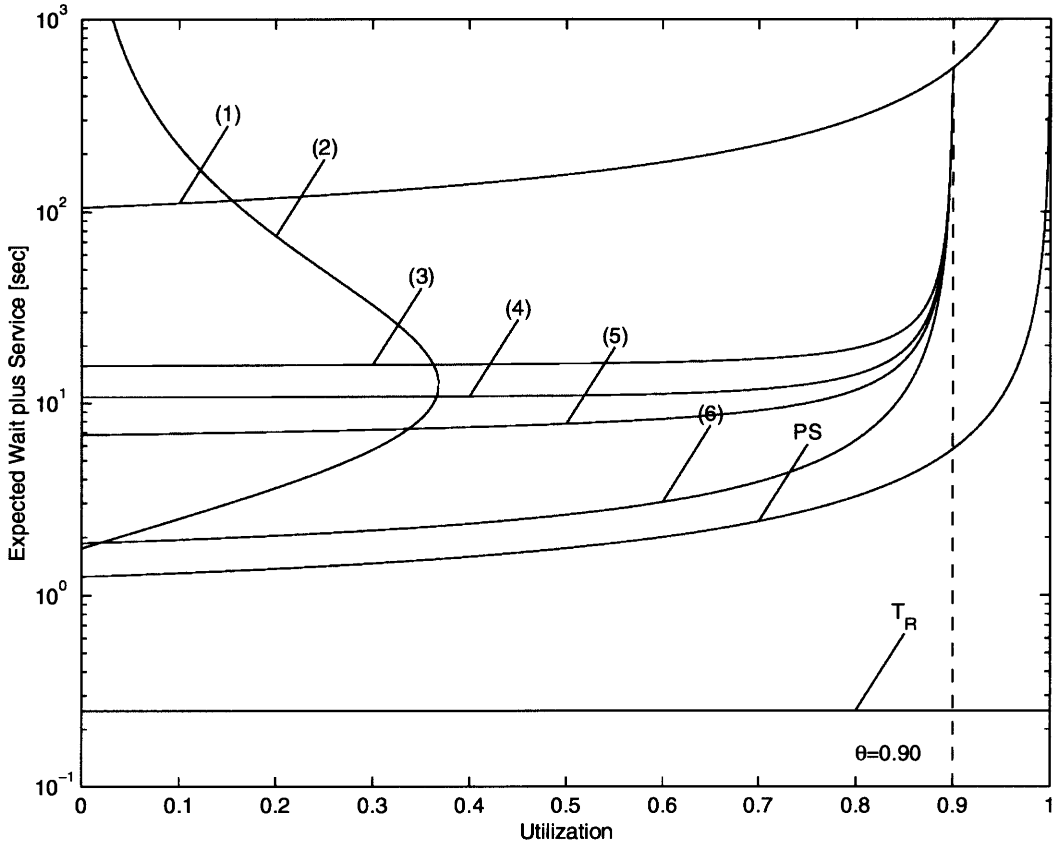


Figure 2.6: $L = 10$ [kb] message transfer for $N = 100$ users at a capacity of $C = 10$ [kbits/sec] with $0 \leq \lambda_c \leq 1$ [msgs/sec]. Note we have assigned 10% of communication resources to placing reservations ($\theta = 0.9$).

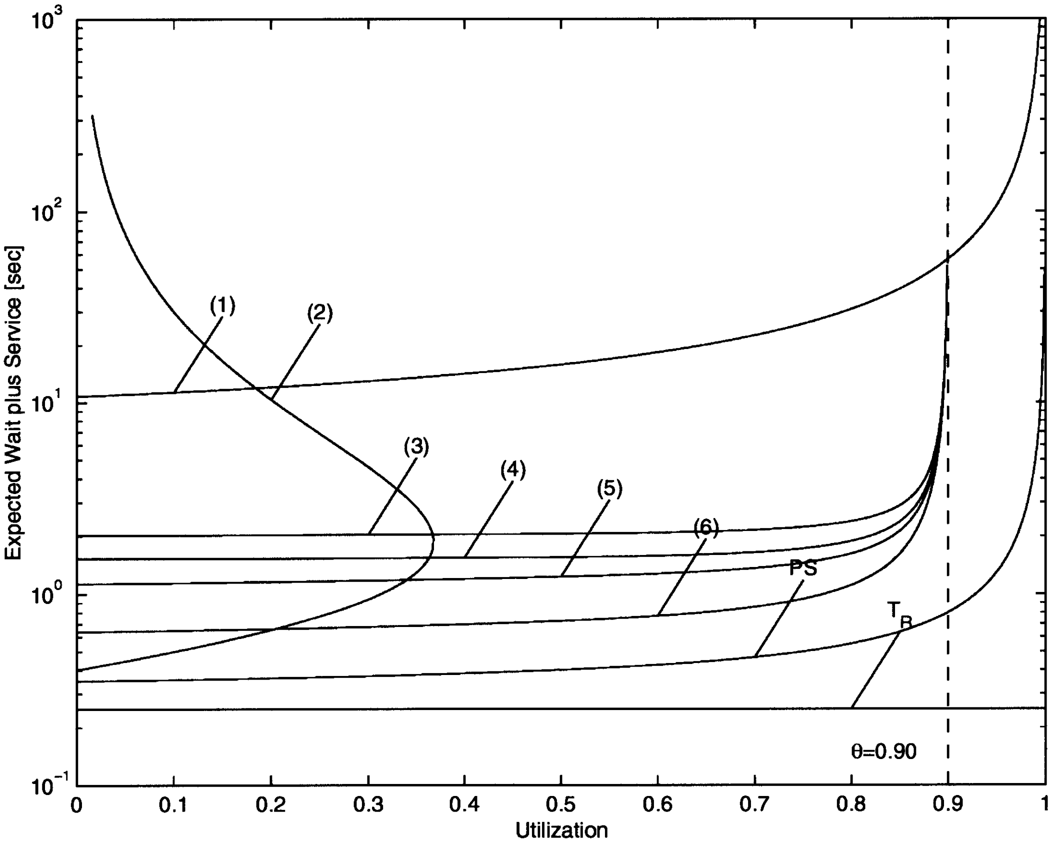


Figure 2.7: $L = 10$ [kb] message transfer for $N = 100$ users at a capacity of $C = 100$ [kbits/sec] with $0 \leq \lambda_c \leq 10$ [msgs/sec]. Note we have assigned 10% of communication resources to placing reservations ($\theta = 0.9$).

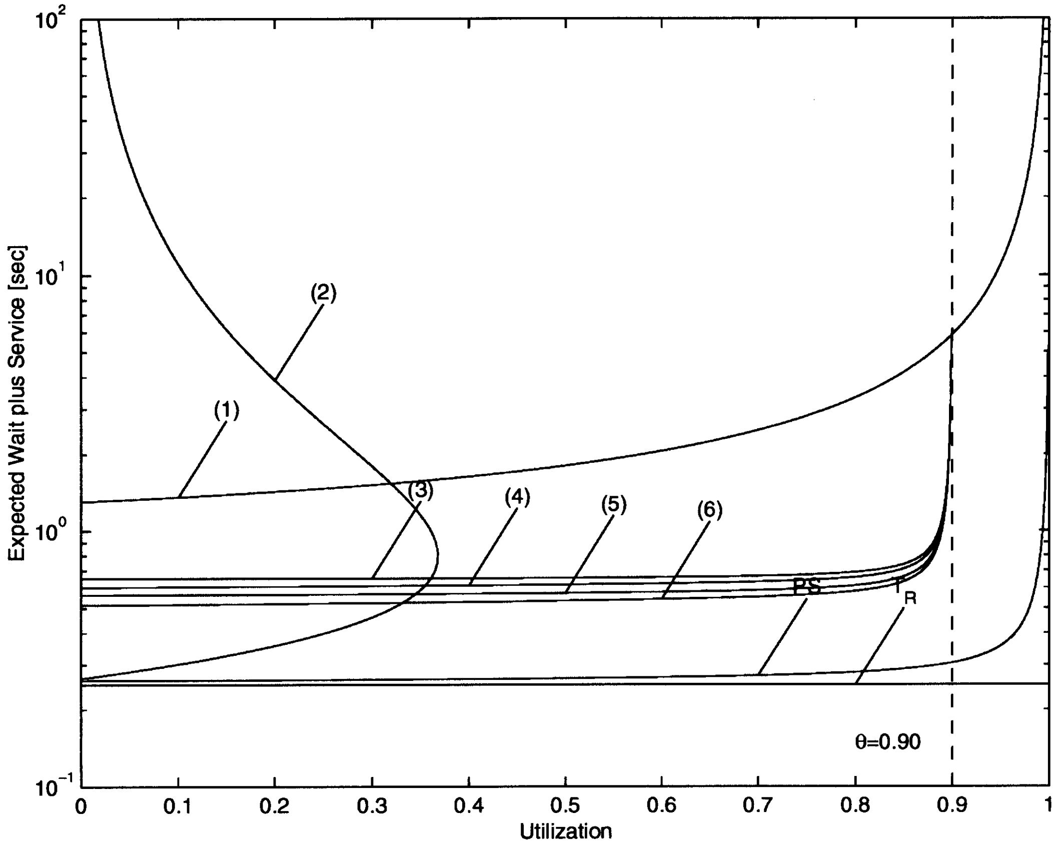


Figure 2.8: $L = 10$ [kb] message transfer for $N = 100$ users at a capacity of $C = 1$ [Mbits/sec] with $0 \leq \lambda_c \leq 100$ [msgs/sec]. Note we have assigned 10% of communication resources to placing reservations ($\theta = 0.9$).

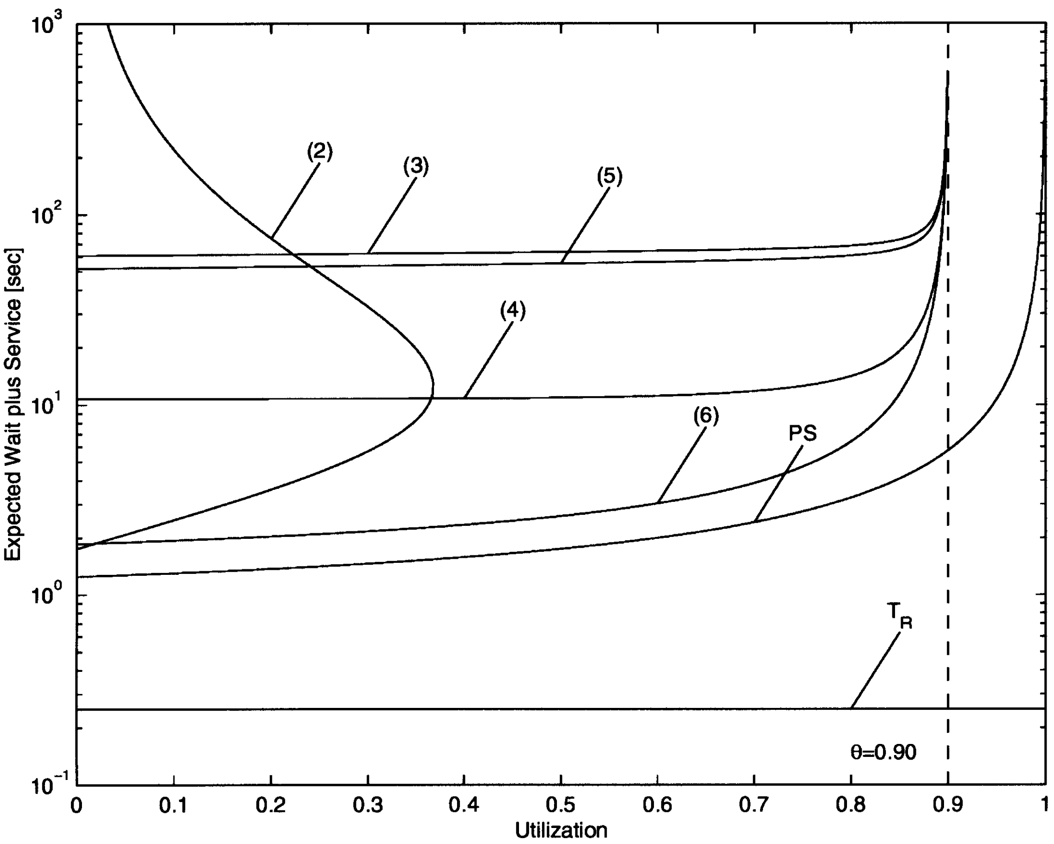


Figure 2.9: $L = 10$ [kb] message transfer for $N = 1000$ users at a capacity of $C = 10$ [kbits/sec] with $0 \leq \lambda_c \leq 1$ [msgs/sec]. Note we have assigned 10% of communication resources to placing reservations ($\theta = 0.9$).

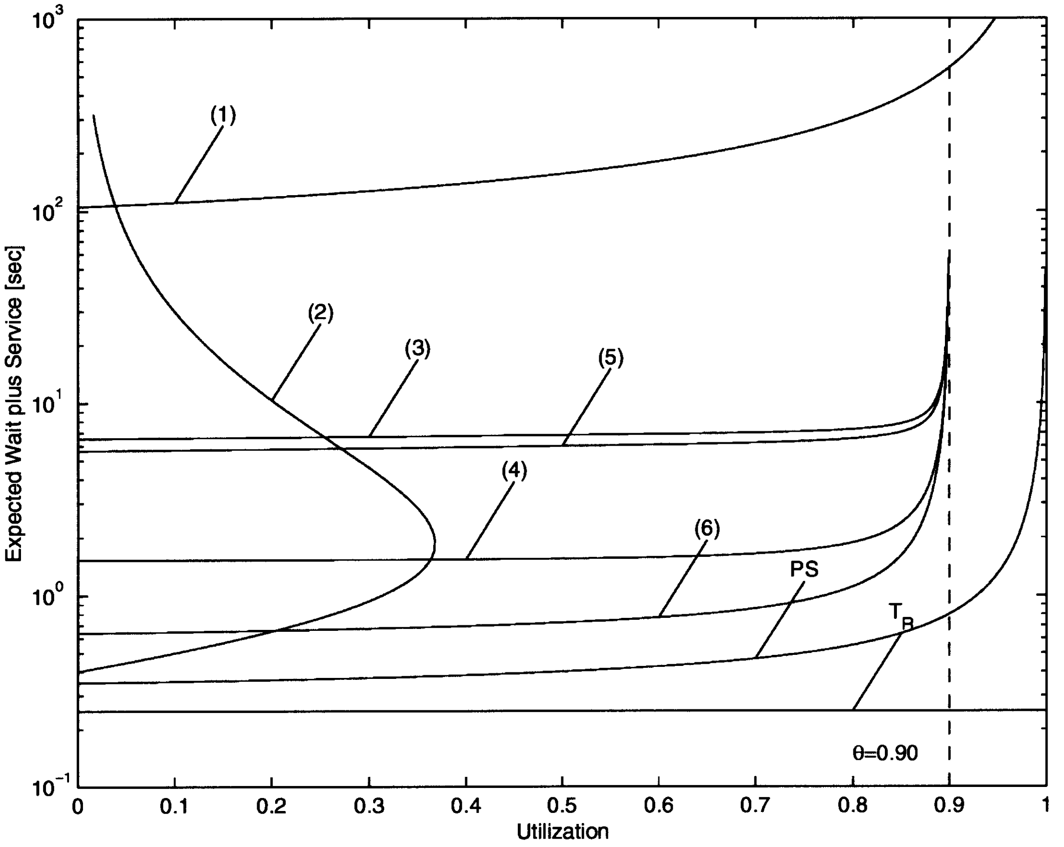


Figure 2.10: $L = 10$ [kb] message transfer for $N = 1000$ users at a capacity of $C = 100$ [kbits/sec] with $0 \leq \lambda_c \leq 10$ [msgs/sec]. Note we have assigned 10% of communication resources to placing reservations ($\theta = 0.9$).

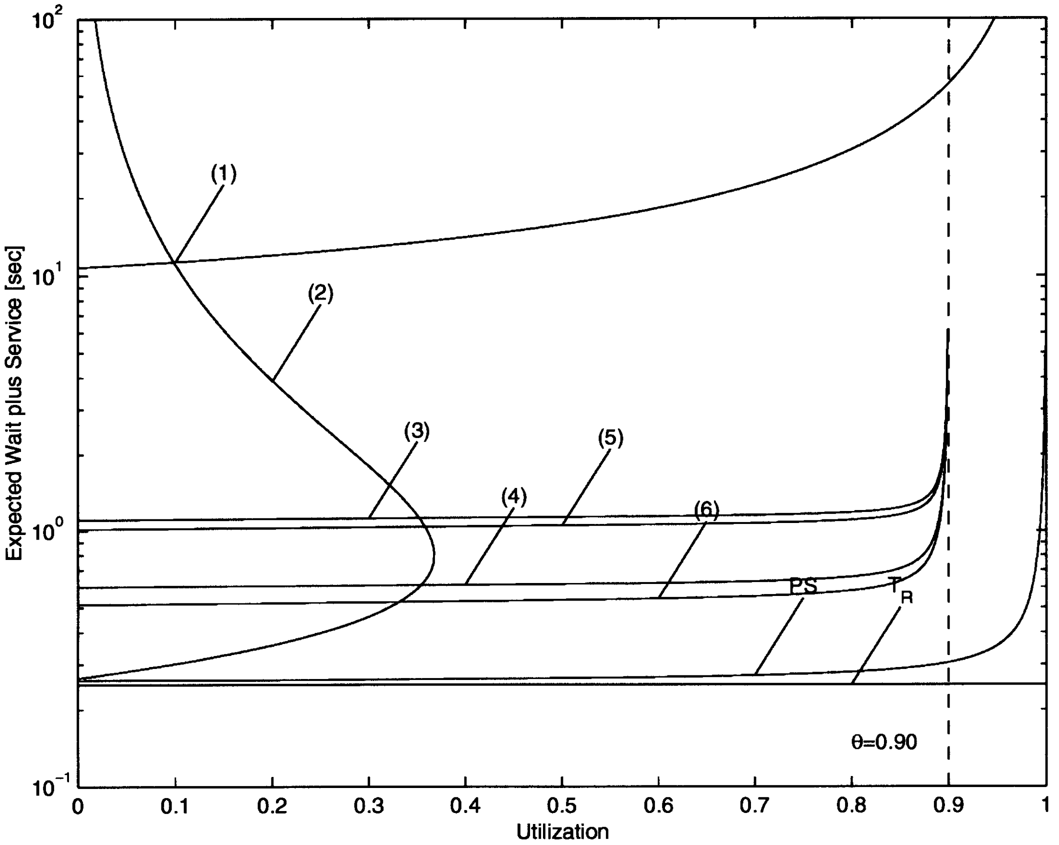


Figure 2.11: $L = 10$ [kb] message transfer for $N = 1000$ users at a capacity of $C = 1$ [Mbits/sec] with $0 \leq \lambda_c \leq 100$ [msgs/sec]. Note we have assigned 10% of communication resources to placing reservations ($\theta = 0.9$).

Channelized Architecture, Random-Access Orderwire

For N small, expected delay shows a small improvement when a random-access orderwire is used with the channelized architecture as opposed to the fixed-access orderwire described above. The distinction between orderwire disciplines becomes more apparent as the number of users N is increased. As N increases, the total frame length of the TDMA orderwire becomes large and becomes a significant term in the expected delay calculation. With the random-access orderwire, however, the infinite-population S-ALOHA approximation remains invariant with N.

Dynamically-Assigned Architecture, Fixed-Access Reservation Channel

The expected delay versus utilization for the dynamically-assigned architecture shows greater variation as utilization is increased. It is noted, however, that expected delay approaches 90% utilization asymptotically, once again supporting intuition with our assignment of $\theta = 0.9$. In general, for the smaller values of N investigated, the dynamically-assigned architecture demonstrates superior expected delay performance over the channelized architecture, regardless of reservation mechanism. As N is increased the TDMA frame length of the reservation channel once again becomes significant. For the case of $N = 1000$ users, the random-access channelized architecture performs better than the fixed-access dynamically-assigned architecture.

Dynamically-Assigned Architecture, Random-Access Reservation Channel

For all parameter values, the dynamically-assigned architecture with a random-access reservation channel demonstrates superior performance. The distinction between the random-access channelized architecture and the random-access dynamically-assigned architecture is exemplified for smaller capacities C. This observation is supported by the intuition developed in Appendix A.2, viz., any channel division of data handling capacity immediately creates a performance inefficiency due to longer queue service times.

2.3.10 Optimized Dynamically-Assigned Architecture

In the above analysis we assumed that 90% of the capacity of the dynamically-assigned uplink was devoted to message transfer with the remaining 10% assigned to the reservation channel, that is, $\theta = 0.9$. This value of θ was chosen such that the amount of capacity assigned to reservations was the same for both the channelized uplink and the dynamically-assigned uplink, allowing a fair analysis between the two. This value of θ, however, is not optimal in the sense of minimizing expected message transmission delay. For a true demonstration of the achievable performance of the dynamically-assigned uplink architecture we must choose θ such that we minimize transmission delay.

The problem, as developed in Appendix B, is of determining the assignment of capacity such that the composite delay experienced with placing a reservation and transmitting a message is minimized, that is,

$$\theta_{\text{opt}} = \operatorname*{argmin}_{\theta \in (0,1)} \left[w_r(\theta) + w_m(\theta) \right]. \tag{2.13}$$

Here $w_r(\theta)$ is the delay associated with using the reservation channel to place a reservation and $w_m(\theta)$ is the delay experienced with queuing and transmitting the message on the data channel. Considering only the random-access reservation channel, θ_{opt} is the optimal capacity assignment which minimizes the S-ALOHA reservation transmission and M/D/1 message queue delays.

With a little thought we realize the capacity assignment θ_{opt} is actually a function of the message arrival rate λ_c. Thus, as utilization changes, optimal delay-minimizing capacity assignment will shift between the reservation channel and the data channel. With the minimization of (2.13) performed in Appendix B, we present Fig. 2.12 as the optimal performance of the dynamically-assigned uplink architecture. As an illustrative example, we choose total uplink capacity of 10 [kbits/sec] and message length of 10 [kbits] as previously investigated. With the same labeling technique as before, curve (4) shows the performance of the channelized uplink

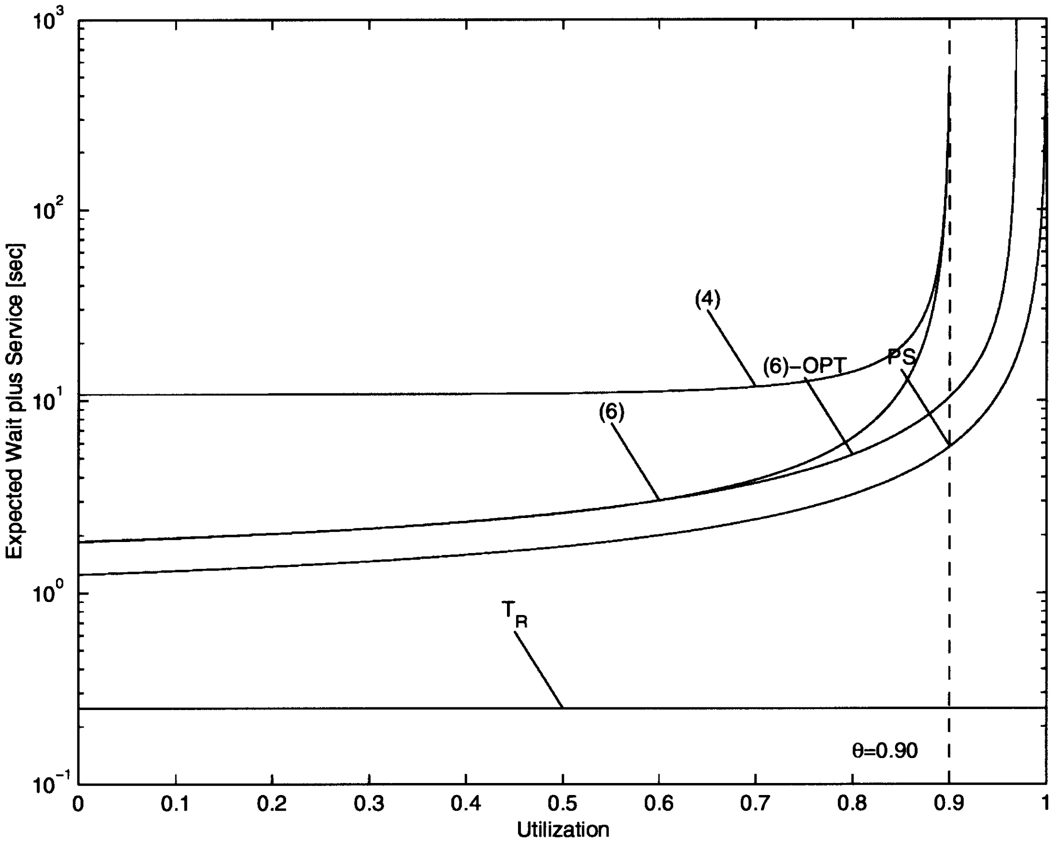


Figure 2.12: Demonstration of the performance of the optimal dynamically-assigned uplink configuration for random-access reservations and $C = 10$ [kbits/sec]. Note allowing θ to change with utilization allows additional service beyond 0.9.

with random-access orderwire and curve (6) shows the performance of the dynamically-assigned uplink with random-access reservation channel and $\theta = 0.9$ for all utilization values, that is, these two curves are the same as in Fig. 2.3.

It is curve (6-OPT) of Fig. 2.12, however, that demonstrates the performance increase which can be realized with the optimal dynamically-assigned uplink if θ is continually adjusted as loading increases. As expected, curve (6-OPT) exhibits the minimum delay for all utilization values, but it is under high utilization conditions where optimizing capacity assignment shows a significant performance increase. Thus the fixed capacity assignment of $\theta = 0.9$ is adequate for low utilization, but is too restrictive at high utilization where the continuously-varying capacity assignment $\theta = \theta_{\text{opt}}(\rho)$ approaching 0.96 offers superior performance.

2.3.11 Channelized Architecture With More Channels

In the above analysis we have assumed the channelized uplink architecture to be composed of $N_c = 10$ uplink channels with one channel assigned as the orderwire. We now wish to investigate channelized uplinks with more than 10 channels and compare their performance to the dynamically-assigned uplink architecture.

In Fig. 2.13 we consider the transfer of a message of length $L = 10$ [kbits] using both architectures. We consider the same demand assignment transmission technique as above, but focus only on the case of fixed-access reservations. We vary the number of channels N_c of the channelized uplink while holding the total uplink capacity C constant at 100 [kbits/sec].

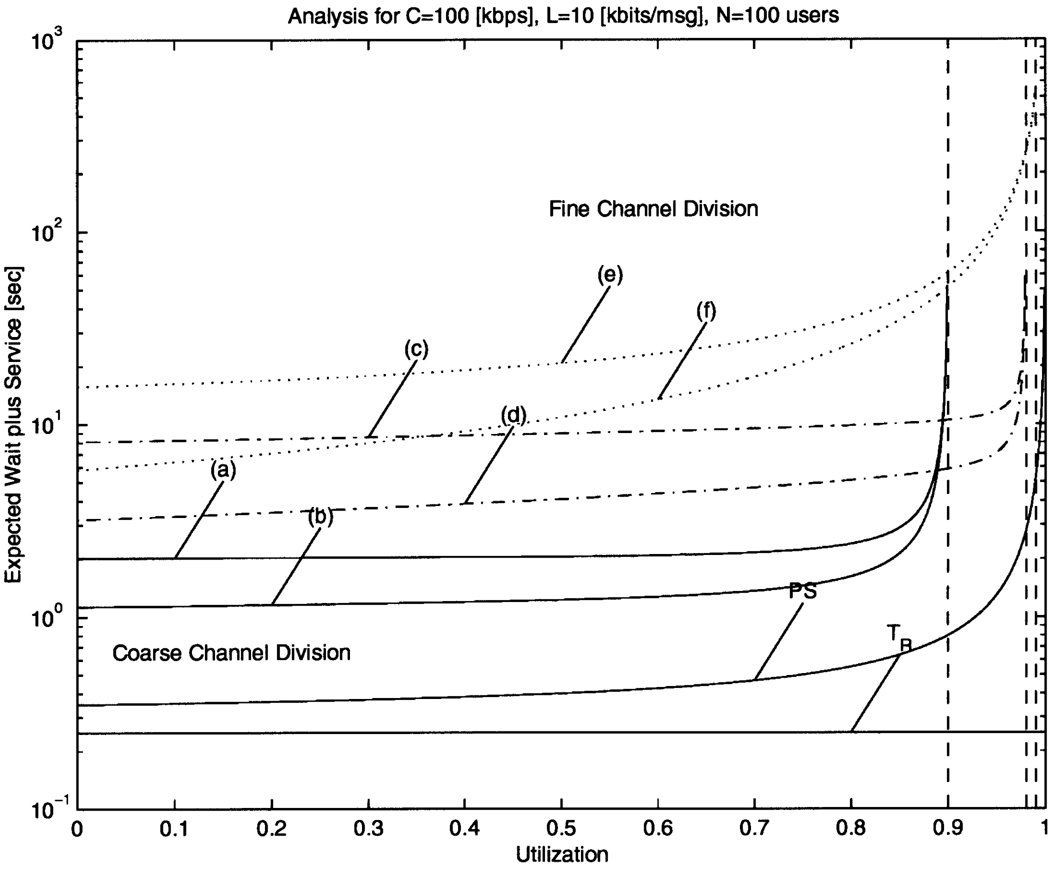


Figure 2.13: Performance of both the channelized and dynamically-assigned uplink architectures for fixed-access reservations and $C = 100$ [kbits/sec] is shown as the channel division N_c of the channelized uplink is increased. Curves (a), (c), and (e) show the performance of the channelized architecture for $N_c \in \{10, 50, 100\}$ channels, respectively. Curves (b), (d), and (f) demonstrate the performance of the dynamically-assigned architecture for $\theta \in \{0.90, 0.98, 0.99\}$.

Curves (a), (c), and (e) of Fig. 2.13 show the performance of the channelized architecture for $N_c \in \{10, 50, 100\}$ channels, respectively, and curves (b), (d), and (f) demonstrate the performance of the dynamically-assigned architecture for $\theta \in \{0.90, 0.98, 0.99\}$.

We see, as N_c increases, we are able to offer higher utilization, but at the price of additional message transfer delay for all utilization values. Additionally, with θ chosen appropriately to compare the channelized uplink to the dynamically-assigned uplink, we see the dynamically-assigned uplink architecture continues to display superior delay performance over the channelized architecture.

It is true that dividing the channelized uplink into more channels *decreases* the expected delay a user experiences before receiving access to a channel. Once the user has the channel, however, the additional delay experienced by transmitting over a channel of smaller capacity significantly *increases* overall message transmission delay.

The problem of determining the optimal number of channels to divide uplink capacity is treated in Appendix A.2. Here it is found that the optimal number of channels N_c to divide uplink capacity into to minimize overall expected message transmission delay is unity. From this result we immediately see the optimality of the dynamically-assigned uplink architecture which offers a *single* high-capacity data channel shared between all users.

2.3.12 Conclusions

The performance of both the channelized uplink architecture and dynamically-assigned uplink architecture has been developed and assessed for the transmission of messages from bursty sources.

To begin, our comparison of the baseline cases of TDMA and S-ALOHA immediately demonstrates the distinction between fixed-access (orthogonal) multiple access transmission and random access transmission. Namely, to achieve high utilization (at the expense of delay) we need to consider orthogonal multiple access techniques. To achieve low delay (at the expense of maximum achievable throughput) we must implement a random access transmission scheme. The process of minimizing delay and simultaneously achieving high throughput requires combining both orthogonal multiple access and random access transmission techniques.

Demand assignment, as investigated for both the channelized and dynamically-assigned uplink architectures, allows us to consider the combination of fixed-access and random access to achieve low delay and high utilization simultaneously. While the access discipline for the data channel is fixed, we considered both architectures using fixed-access and random-access reservations. The performance of using fixed-access reservations is dependent upon the number of users N and we see, for $N \leq 100$, the dynamically-assigned uplink continually displays superior performance over the channelized uplink, regardless of reservation discipline. For $N > 100$, on the other hand, the performance of random-access reservations shows superior performance for both architectures over that of fixed-access reservations. Of all cases considered, it is only the dynamically-assigned uplink architecture with random-access reservations that shows consistently superior performance.

Further praise for the dynamically-assigned uplink architecture with random-access reservations is developed when we consider optimizing the capacity assignment between the reservation channel and the data channel. A reconfigurable uplink architecture, where capacity can be traded between reservations and message transfer as utilization increases, allows us to come even closer to the perfect scheduling results at high loading. The fixed boundaries of the channelized architecture, however, constrain the flexibility of capacity assignment and is less optimal.

Finally we consider the idea of creating more channels for the channelized architecture while holding total uplink capacity constant. We found this *not* to be beneficial as the expected message delay increased with the number of channel capacity divisions. This result, from a queuing theory perspective, is not surprising since any division of channel capacity into separate sub-channels is non-optimal as discussed in Appendix A.2.

2.4 Client-Server Session Performance

2.4.1 Overview

Observation of current trends of communication requirements points to the need of supporting users' ability to browse large repositories of information in real-time. One technique to offer this service would be to develop some large capacity broadcast downlink to all users. In this scenario, a user's terminal would capture only the information which may be of relevance for later perusal by the user. The difficulty with this scheme, however, is that large amounts of information storage are required at each user terminal and the flexibility to instantly recall missed broadcasts is difficult.

A technique to affect interactive information browsing and retrieval is to allow each user's terminal to specifically request documents from the repository. In this way a *client* application resides at the terminal and, upon user demand, accesses some information *server*. A general characteristic of client-server interaction is the asymmetry of transmission requirements from client to server and server to client. The user's client application generally sends small information request packets to the server. In response to these requests the server responds with information many orders of magnitude greater than the original request packet. For example, a user operating a WWW browser[3] (the client) sends small URL[4] requests to the server. The server responds by offering a content-rich HTML[5] document, possibly with multimedia information. In this example it is easy to identify the asymmetric nature of transmission requirements.

The challenge is to offer interactive client-server communication in an efficient and responsive manner. The period in which a client and server are able to communicate is termed the ***session*** and, for the WWW browser example above, would constitute the time the user is actively using the browser to retrieve information. Since many clients access only a few servers, it is necessary to share uplink communication resources in an intelligent way allowing many simultaneous sessions and thus maximizing the user capacity of the system.

In this section we investigate the channelized and dynamically-assigned uplink architecture performance for handling client-server interaction. A general model for a client-server session is offered and the performance of both candidate architectures is developed. The three metrics of evaluation include:

Session Establishment Delay The mean delay experienced when a user wishes to initiate a session between client and server.

Packet Transmission Delay Once the session is established we must assess the mean delay experienced by the user's request packets.

Maximum Number of Simultaneous Sessions This a measure of the user capacity of the system, that is, the maximum number of sessions that can be active at the same time.

Perfect scheduling results are considered for assessing superior performance between the channelized and dynamically-assigned uplink for the three parameters presented above. To further develop intuition, real-world parameter assumptions are made to demonstrate the operating regions of superior architecture performance.

2.4.2 Session Model

We model a typical client-server interaction by decomposing the problem into two parts: session statistics and inter-session packet arrival statistics. Sessions are assumed to have exponentially distributed lengths with mean T. Additionally, sessions are assumed to be arriving from an infinite population of users according to a Poisson process of rate λ_s. Within the session, packets with mean length τ arrive according to a Poisson process of rate λ_p. Fig. 2.14 shows a typical session arrival with inter-session packet arrivals. Thus the traffic model for a single user is the ***interrupted Poisson process***[6] with rate λ_p. It is important to note the session length T is independent of the packet arrival statistics and transmission capacity, an assumption that is valid when considering short packets from client to server.

2.4.3 Channelized Uplink Architecture

Our analysis for the channelized uplink architecture is fairly straightforward–common results for the M/M/k queue offer us the desired delay statistics of interest. Specifically, for an channelized uplink with total capacity C divided into N_c channels, we can determine the expected wait for an available channel by considering the uplink as an M/M/N_c queue with exponential arrivals of mean length T arriving according to a Poisson process of mean rate λ_s. The expected wait to

[3] The world wide web (WWW) is an information retrieval service available mainly on the Internet consisting of interconnected hypertext documents. A WWW browser is a software application operating on a user's computer. With the browser a user is able to retrieve hypertext documents and navigate to other documents on the network.

[4] The documents on the WWW are accessed by specifying their address. The Universal Resource Locator (URL) is the specific address identifying both the WWW server and the location of the document within the server.

[5] The HyperText Markup Language (HTML) is a scripting language that allows for the integration of hypertext document links and rich multimedia content with a simple text file.

[6] The interrupted Poisson arrival process is discussed in Chapter 3.

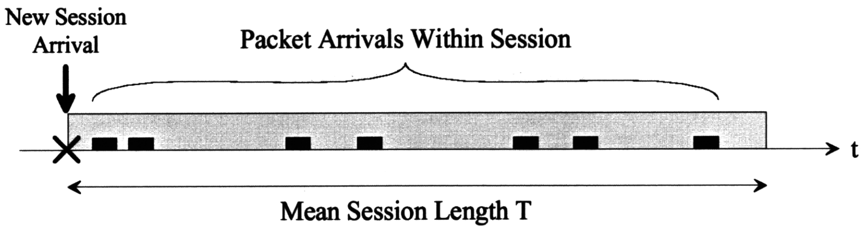


Figure 2.14: A typical session arrival is shown with mean duration T [sec].

begin our session is the expected wait for the first-available channel. Thus, on average, a user's client application must wait

$$\mathcal{W}_{\mathrm{S,FDM}} = \mathcal{Q}_{\mathrm{M/M/k}}\left(\lambda_s, \frac{1}{T}, N_c\right) \tag{2.14}$$

[sec] before initiating a session with the server. Once the session is established, the user has exclusive access to the assigned channel during the entire session duration. The delay per packet a given user experiences within the session is associated only with prior packet arrivals from the user, that is, the delay associated with an M/M/1 queue. For a given user, the expected delay for transmitting a packet of mean length l_p [bits] within the session is

$$\mathcal{W}_{\mathrm{P,FDM}} = \mathcal{Q}_{\mathrm{M/M/1}}\left(\lambda_p, \frac{C}{l_p N_c}\right) + \frac{l_p N_c}{C} \tag{2.15}$$

[sec] where we have ignored the constant propagation delay associated with transmission.

2.4.4 Dynamically-Assigned Uplink Architecture

Analysis of the dynamically-assigned uplink architecture is a bit more complicated. Here we must aggregate the packets from all active sessions onto one channel, that is, the traffic on the channel is Poisson with a rate which is proportional to the number of active sessions. In this scenario, the traffic model of all users is a ***Markov-modulated Poisson process*** (MMPP). The MMPP is a doubly stochastic process which generates Poisson arrivals with an arrival rate which is a function of some continuous-time, finite-state Markov process. The MMPP in its general form is presented in Chapter 3. For our purposes, since the packet arrival rate within each session is the same for all users, we may resort to a computationally simpler technique for obtaining the delay and utilization statistics for the dynamically-assigned architecture.

Once again we are able to decompose the analysis into session arrivals and packet arrivals. We use an M/M/∞ queue to determine the expected number of simultaneous sessions by imagining a single server of the M/M/∞ will be assigned to service only one session. For any M/M/∞ queue with mean customer arrival rate λ and mean service rate μ, the expected number of busy servers N is given by

$$N = \frac{\lambda}{\mu}. \tag{2.16}$$

This allows us to realize the expected number of active sessions N_s in our model to be

$$N_s = \lambda_s T. \tag{2.17}$$

Note there is no waiting associated with an M/M/∞ queue since each arriving customer is immediately assigned a new server and receives exceptional service. Now that we know the expected number of active sessions, the mean aggregate packet arrival rate is simply

$$\lambda_a = N_s \lambda_p. \tag{2.18}$$

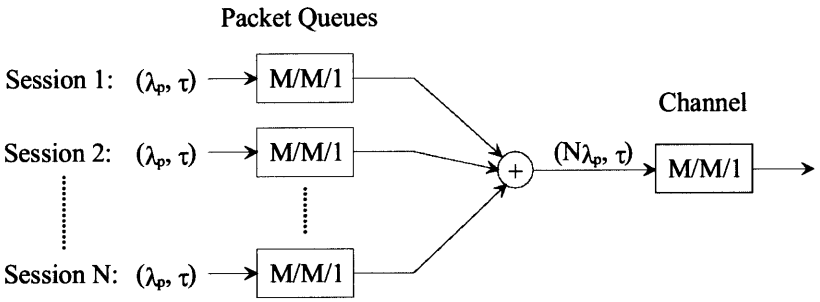

Packet Queues
Session 1: (λ_p, τ)
Session 2: (λ_p, τ)
Session N: (λ_p, τ)
M/M/1
M/M/1
M/M/1
+
$(N\lambda_p, \tau)$
Channel
M/M/1

Figure 2.15: The packet arrivals from all active sessions are aggregated into the single data channel queue.

This aggregate arrival rate is offered to the dynamically-assigned uplink channel. Delay and utilization statistics are calculated by modeling the aggregate channel as an M/M/1 queue. Previously, for the channelized uplink, it was necessary to wait for a channel to become available, generally a considerably large wait under high utilization. With the dynamically-assigned uplink, however, we only need to wait for the single high-rate channel servicing packets to become available before we start our session. The expected wait to start a session with the dynamically-assigned architecture is

$$\mathcal{W}_{\mathrm{S,TDM}} = \mathcal{Q}_{\mathrm{M/M/1}}\left(N_s\lambda_p, \frac{C}{l_p}\right) \tag{2.19}$$

[sec] where, again, we assume data packets are l_p [bits] in length and our total uplink capacity is C [bits/sec].

We now focus on the delay a packet experiences within a session. The dynamically-assigned uplink requires special attention since the channel is shared amongst all users on a per-packet bases. It is necessary to realize a user's data packets experience two queuing delays before reaching their destination. The first delay is associated with the queuing of packet arrivals from a given user, a delay that is easily obtained from M/M/1 queue analysis. These departures from the user's packet queue then enter the channel which, again, is modeled as an M/M/1 queue receiving arrivals from all active sessions. Thus we have a cascade of queues, as shown in Fig. 2.15, with many M/M/1 user packet queues aggregated into the single M/M/1 channel queue.

For arbitrary arrival and service processes, the analysis of such cascaded queue arrangements requires special attention since common analysis assumes Poisson arrivals into each queue, a property which cannot always be guaranteed with general service disciplines. With cascaded M/M/k queues, however, *Burke's Theorem* states the departure process of the i^{th} queue output is Poisson with the same mean rate as the queue input process[13]. Additionally, each departure process of the M/M/k queue is independent of the departure processes of the $k-1$ remaining queue outputs. Thus, with the saving grace of Burke's Theorem, we are able to decompose the problem and treat the two queues as being independent. Therefore the mean transmission delay a user's packet experiences within a session is

$$\mathcal{W}_{\mathrm{P,TDM}} = \mathcal{Q}_{\mathrm{M/M/1}}\left(\lambda_p, \frac{C}{l_p}\right) + \mathcal{Q}_{\mathrm{M/M/1}}\left(N_s\lambda_p, \frac{C}{l_p}\right) + \frac{l_p}{C} \tag{2.20}$$

[sec] for perfect scheduling ignoring propagation delay.

2.4.5 Model Parameters

The following table quickly summarizes the parameters of our model.

Parameter	Description	Units
C	Total uplink channel capacity	[bits/sec]
λ_s	Session mean arrival rate	[sessions/sec]
T	Mean of exponentially distributed session length	[sec]
λ_p	Packet mean arrival rate within a session	[packets/sec]
l_p	Packet length	[bits]
τ	Packet length when transmitted at rate C	[sec]
N_c	Number of channels for channelized arch.	[channels]
$\mathcal{W}_{\mathrm{S,FDM}}$	Mean session wait time for channelized arch.	[sec]
$\mathcal{W}_{\mathrm{P,FDM}}$	Mean packet transmission time for channelized arch.	[sec]
N_s	Mean num. of concurrent sessions for dynamically-assigned arch.	[sessions]
$\mathcal{W}_{\mathrm{S,TDM}}$	Mean session wait time for dynamically-assigned arch.	[sec]
$\mathcal{W}_{\mathrm{P,TDM}}$	Mean packet transmission time for dynamically-assigned arch.	[sec]

2.4.6 Assumed Parameter Values

Client to server transmission delay is investigated for both the channelized and dynamically-assigned uplink architectures while varying the four parameters of our model throughout real-world values. Session arrival rate λ_s is varied from one session arrival every 1000 [sec] to one session arrival every 10 [sec]. The mean session length T, being exponentially distributed, is investigated to a maximum of 60 [minutes].

The packet arrival process, following from the nature of the client to server interaction, is a sporadic source of short packets. The packet arrival rate λ_p within the session is varied from one packet every 100 [sec] to 10 packets per [sec]. The packets are assumed to be transmitted in $\tau = 10^{-3}$ [sec] for the dynamically-assigned uplink and $N_c\tau$ [sec] for the channelized uplink.

2.4.7 Analysis

Session Establishment Delay

To begin the evaluation of client to server data transmission we consider the expected delay experienced when a user wishes to initiate a session with the server. For the example of the WWW browser, we are interested in determining the delay from when the user first starts the browser to when communication with the HTML server can begin. For the channelized uplink architecture, this delay manifests itself in the expected wait for one of the N_c data channels to become available. For the dynamically-assigned uplink architecture, on the other hand, the expected delay until a session can be initiated is a function of the utilization of the data channel, i.e., both the number of active sessions and their packet arrival rates.

Instead of producing multiple delay vs. utilization figures for both architectures varying our four session parameters, we attempt to summarize the operating regions where either the channelized or dynamically-assigned uplink demonstrates superior performance. That is, for the instance of delay, we characterize the parameter values where use of the channelized uplink produces smaller delay than the dynamically-assigned uplink or where the use of the dynamically-assigned uplink produces smaller delay than the channelized uplink.

Figs. 2.16-2.19 demonstrate the operating regions where either the channelized uplink or dynamically-assigned uplink shows superior session establishment delay. For the parameters T and λ_s, the shaded region identifies the operating points where the session establishment delay of the dynamically-assigned uplink is less than the session establishment delay of the channelized uplink. Four figures are shown as the packet arrival rate λ_p is increased from 10^{-2} [packets/sec] to 10 [packets/sec].

Packet Transmission Delay

Our second measure of architecture performance is the expected delay to convey a packet from client to server once the session is established. Again, we determine the superior architecture in

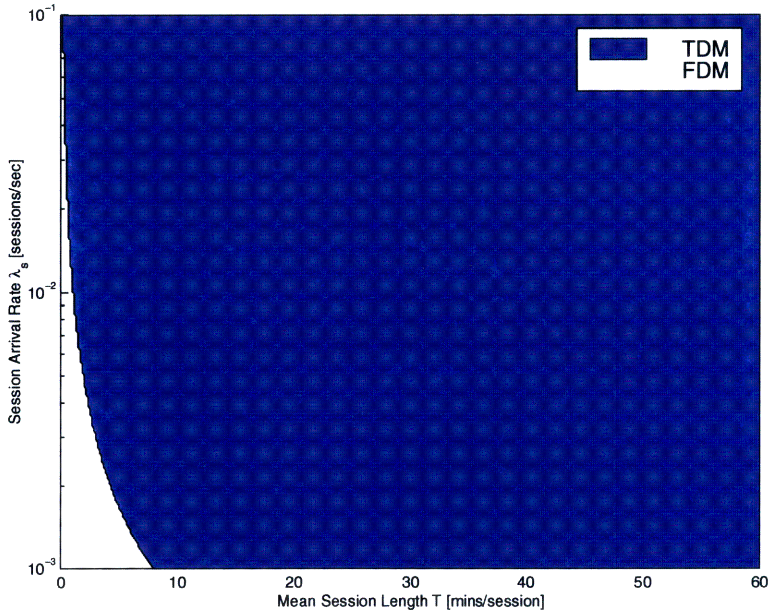


Figure 2.16: Summary of superior architecture operating points for minimizing session establishment delay assuming $\lambda_p = 10^{-2}$ [packets/sec] and $\tau = 10^{-3}$ [sec].

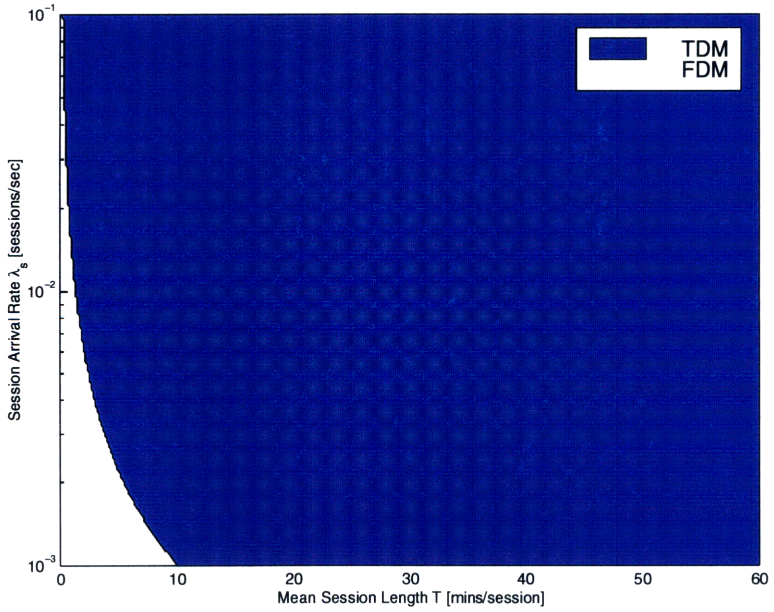


Figure 2.17: Summary of superior architecture operating points for minimizing session establishment delay assuming $\lambda_p = 10^{-1}$ [packets/sec] and $\tau = 10^{-3}$ [sec].

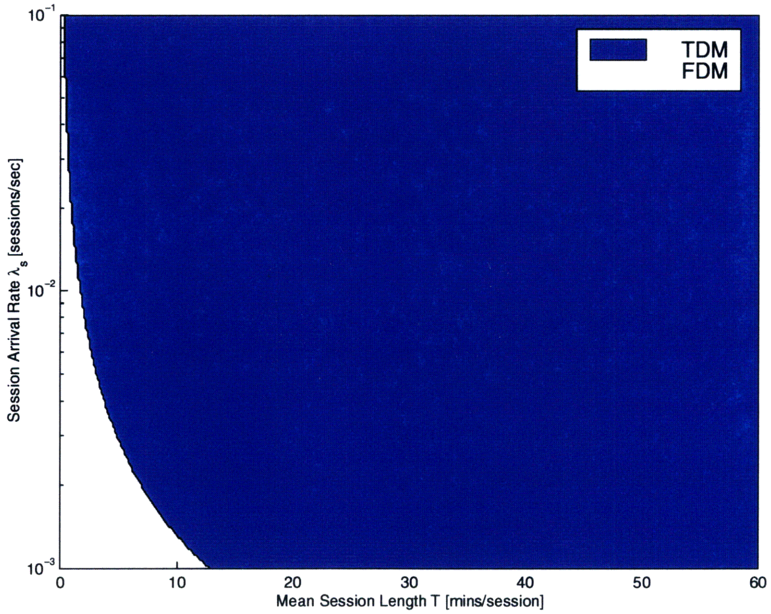


Figure 2.18: Summary of superior architecture operating points for minimizing session establishment delay assuming $\lambda_p = 1$ [packet/sec] and $\tau = 10^{-3}$ [sec].

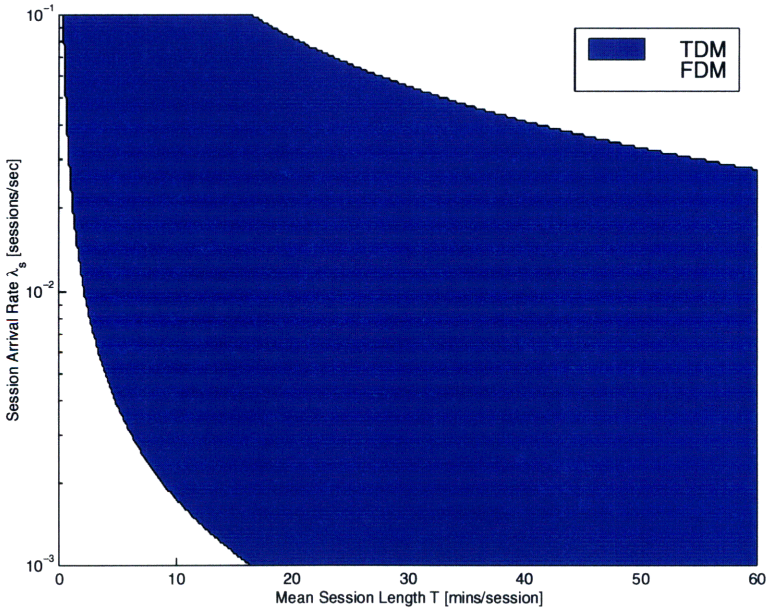


Figure 2.19: Summary of superior architecture operating points for minimizing session establishment delay assuming $\lambda_p = 10$ [packets/sec] and $\tau = 10^{-3}$ [sec].

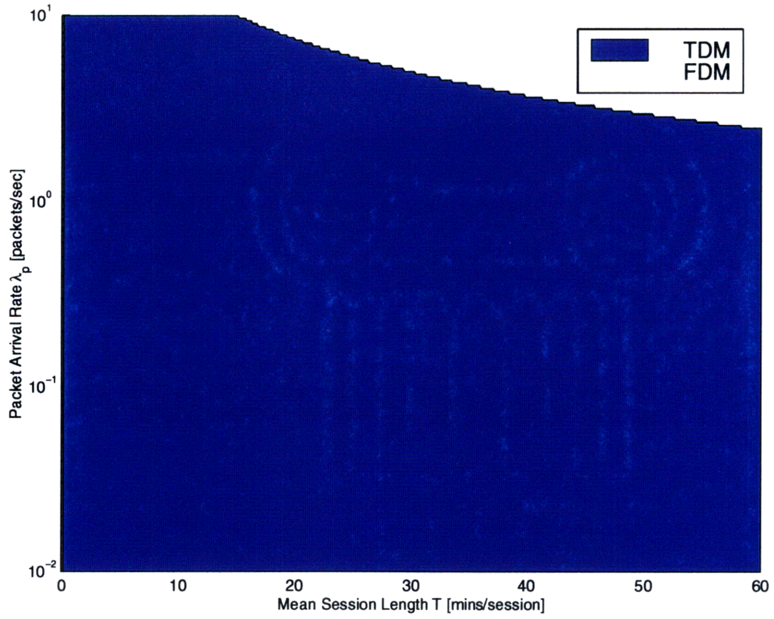


Figure 2.20: Summary of superior architecture operating points for minimizing packet transmission delay assuming $\lambda_s = 10^{-1}$ [sessions/sec] and $\tau = 10^{-3}$ [sec].

terms of packet transmission delay with the rule

$$\mathcal{W}_{\text{P,FDM}} \quad \underset{\text{FDM}}{\overset{\text{TDM}}{\gtrless}} \quad \mathcal{W}_{\text{P,TDM}}. \tag{2.21}$$

That is, if we define

$$\alpha = \frac{\mathcal{W}_{\text{P,FDM}}}{\mathcal{W}_{\text{P,TDM}}} = \frac{\mathcal{Q}_{\text{M/M/1}}\left(\lambda_p, \frac{1}{N_c\tau}\right) + N_c\tau}{\mathcal{Q}_{\text{M/M/1}}\left(\lambda_p, \frac{1}{\tau}\right) + \mathcal{Q}_{\text{M/M/1}}\left(N_s\lambda_p, \frac{1}{\tau}\right) + \tau} \tag{2.22}$$

then, in terms of packet transmission delay, we declare the channelized uplink architecture to be superior to the dynamically-assigned uplink architecture when $\alpha < 1$ and, consequently, the dynamically-assigned uplink to be superior to the channelized uplink when $\alpha > 1$, viz.,

$$\alpha \quad \underset{\text{FDM}}{\overset{\text{TDM}}{\gtrless}} \quad 1. \tag{2.23}$$

While investigating the packet transmission delay for the values of λ_s, T, λ_p, and τ of interest to us in this investigation, we find only the variation of λ_p and T for $\lambda_s = 10^{-1}$ [sessions/sec] produces major variation of α as expressed by (2.22). Thus Fig. 2.20 shows the regions of λ_p and T where α shows variation about unity.

Maximum Number of Simultaneous Sessions

A third major distinction between the channelized and dynamically-assigned uplink architectures is the maximum number of simultaneous sessions that are supported. That is, for given parameters, which architecture allows for more client-server interactions to take place concurrently? To answer this question we must first consider the mechanisms within each candidate architecture which constrain the number of active sessions.

For the channelized uplink, clearly the number of available uplink channels limits the number of active sessions. Since only one user may utilize an uplink channel for transmission, the maximum number of active sessions is N_c. For the dynamically-assigned architecture, we note the

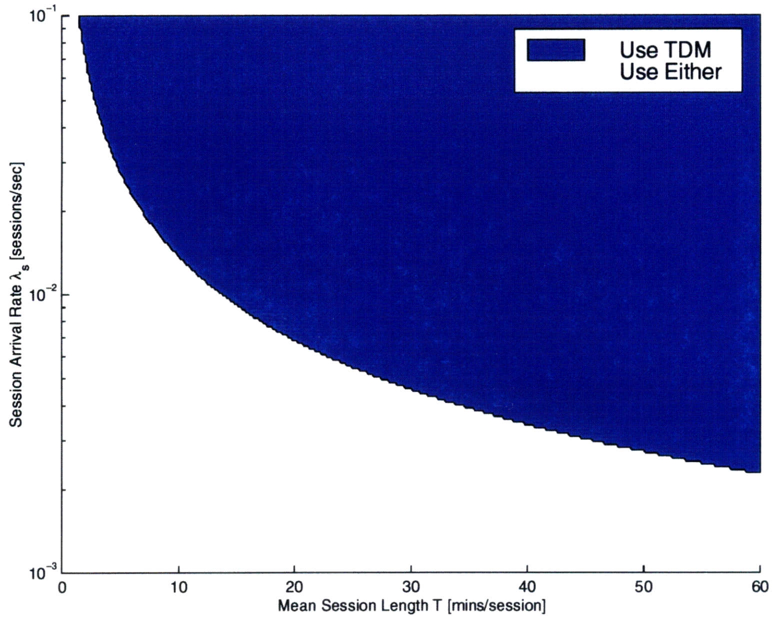


Figure 2.21: Summary of superior architecture operating points maximizing number of simultaneous sessions assuming $\lambda_p = 1$ [packets/sec] and $\tau = 10^{-3}$ [sec]. The shaded region marks the parameter values where, on average, all N_c uplink channels of the channelized uplink are occupied.

ability of the data channel to handle packet arrivals limits the number of active sessions. When the utilization of the data channel is maximum, the dynamically-assigned uplink architecture cannot accept any new sessions. Due to the stochastic nature of the data channel utilization we can only determine the expected number of simultaneous sessions of mean length T [sec] supported by the dynamically-assigned uplink to be

$$N_{s,max,T} = \frac{1}{\lambda_p \tau} \tag{2.24}$$

corresponding to the instance when the data channel is fully loaded.

Again, instead of presenting a quantitative comparison of number of simultaneous sessions, we instead determine the regions where the dynamically-assigned uplink will allow *more* concurrent sessions than the channelized uplink. The shaded region of Fig. 2.21 demonstrates where, on average, all N_c data channels of the channelized uplink are occupied and where the dynamically-assigned architecture has capacity for more concurrent sessions beyond N_c. In the non-shaded region, however, either uplink architecture can be used since for these values of T and λ_s, on average, the channelized uplink has data channels available for immediate use.

2.4.8 Interpretation of Results

Session Establishment Delay

Interpreting Figs. 2.16-2.19 we see, for the majority of the parameter values we are interested in, the dynamically-assigned uplink architecture offers the smallest delay for establishing a new session. It is important to note, however, the channelized architecture does offer superior session establishment performance when we have short, infrequent sessions. In this instance, on average, the channelized architecture has channels available for immediate use by new session arrivals, that is, the channelized architecture offers *exceptional service* to short, infrequent session arrivals.

The four plots offered show the variation of session establishment delay as the packet arrival rate λ_p is increased. We see, for $\lambda_p \leq 1$ [packet/second], the dynamically-assigned uplink architecture remains superior for most parameter values, but the dynamically-assigned uplink is becoming less effective for offering small establishment delay for short sessions. For $\lambda_p > 1$ the channelized uplink offers smaller session establishment delay for long-duration, frequently-occurring sessions in addition to short, sporadic session arrivals. As noted in the upper region of Fig. 2.19, the loading of the dynamically-assigned uplink data channel is maximum and the delay for access is infinite. The expected delay for an available channel for the channelized uplink, albeit large, is still finite.

One would think that as session length T and session arrival rate λ_s increase, the session delay performance of both the channelized and dynamically-assigned architectures would be the same. This is *not* the case, however, since the delay associated with session establishment for the channelized uplink is a function of T and λ_s whereas the delay associated with session establishment for the dynamically-assigned uplink is not only dependent upon T and λ_s, but also λ_p and τ. Hence the delay experienced with session establishment is independent of the inter-session traffic statistics (i.e. λ_p and τ) for the channelized uplink architecture, but not for the dynamically-assigned uplink architecture.

Packet Transmission Delay

The results obtained while investigating the expected packet transmission delay are surprising in the sense that, for most parameter values, the *shared* dynamically-assigned uplink channel offers smaller delay than the single *exclusive-use* channel assignment of the channelized uplink architecture. From (2.22) we see why this is. The utilization of the shared dynamically-assigned uplink channel is given by

$$\rho_{\text{TDM}} = N_s \lambda_p \tau \tag{2.25}$$

and substituting for the expected number of active sessions we have

$$\rho_{\text{TDM}} = \lambda_s T \lambda_p \tau. \tag{2.26}$$

For a lightly loaded channel, that is $\rho_{\text{TDM}} \ll 1$, the delay associated with channel activity of other active sessions is small and α from (2.22) tends towards N_c. Thus the dynamically-assigned uplink architecture offers smaller message transmission delay because it is able to service packet arrivals a factor of N_c times faster than the channelized uplink architecture. When the channel is heavily loaded with $\rho_{\text{TDM}} \to 1$, the channel activity of other users of the dynamically-assigned uplink results in delay greater than the expanded service time of the channelized uplink channel. This is demonstrated in Fig. 2.20 where for λ_s, λ_p, and T large we see the channelized uplink architecture offers smaller overall packet transmission delay.

Maximum Number of Simultaneous Sessions

Fig. 2.21 demonstrates the region where, on average, all N_c data channels of the channelized uplink will be occupied and, consequently, where the dynamically-assigned uplink architecture should be utilized. Again, since the channel availability of the channelized architecture is independent of the inter-session traffic statistics, the boundary shown in Fig. 2.21 is valid for all λ_p and τ. Note, however, the shaded region of Fig. 2.21 is *not* implying optimality of the dynamically-assigned uplink architecture for all λ_p. Proper comparison would involve investigating the intersection of the shaded regions of Figs. 2.16-2.19 and Fig. 2.21.

2.4.9 Conclusions

We have developed techniques to asses the performance of the channelized and dynamically-assigned uplink architectures for handling traffic generated from client-server interaction. For typical parameters values of interest we have identified the session establishment delay, packet transmission delay, and maximum number of concurrent sessions as metrics to evaluate the performance of both architectures.

Considering these three metrics, our results show for mean session lengths less than 10 [mins] and very light loading, on average, the channelized architecture will allow for the quickest session establishment, but is suboptimal in the sense of delay per packet during the length of the session. For sessions of greater length and for heavy loading, however, the dynamically-assigned uplink architecture demonstrates both superior session establishment delay and superior per packet delay once the session is initiated.

While considering the maximum number of simultaneous sessions, it is found either architecture will support the session demand for light loading, but it is only the dynamically-assigned uplink that can support more than N_c concurrent sessions. Therefore it is the dynamically-assigned uplink which allows the most number of users to simultaneous utilize the communication resources offered by the satellite system.

The concept of modeling sessions as opposed to just single message transfer arises when we consider sources which are bursty. In this section we have shown how poorly the channelized architecture supports client-server applications. This poor performance stems from the fact that once a user occupies a channel, they hold the channel during the whole duration of the session, even when no useful information is being conveyed. The dynamically-assigned architecture, however, allows the fixed capacity to be assigned to the session when it has information to convey, and is statistically multiplexed between other sessions when idle. In addition, the dynamically-assigned architecture instantaneously assigns all available capacity to a single user thus allowing extremely fast packet transmission.

2.5 Summary

This chapter presents two distinctly different communication architectures and develops techniques for assessing their performance for handling two diverse data communication requirements. The channelized uplink architecture is characteristic of existing systems which have been developed for offering narrow-band voice services, whereas the dynamically-assigned uplink architecture is a proposed next-generation satellite uplink architecture to handle data communication requirements from bursty packet sources.

The following list highlights the major points of this chapter:

- Demand assignment allows us to combine both orthogonal multiple access and random access into one transmission scheme. Short reservation packets benefit from the low delay of random access while large messages are transmitted contention-free and with high capacity utilization with orthogonal multiple access.

- The dynamically-assigned uplink offers the full transmission capacity to any given user, regardless of system loading.

- The dynamically-assigned uplink architecture can take advantage of reconfigurable uplink architectures by dynamically allocating uplink capacity between the reservation channel and data channel.

- For all parameter values investigated, the dynamically-assigned uplink architecture offers the lowest expected message transmission delay, approaching that of the perfect scheduling lower bound on delay.

- The channelized uplink architecture is inherently non-optimal for the transmission of large messages due to the small fraction of total uplink capacity assigned to any given user.

- Finer division of the channelized uplink into more channels moves us further away from the perfect scheduling bound on expected message transmission delay.

- Future uplink architectures must consider the transmission characteristics of the ubiquitous client-server model and tailor communication resource assignment to affect efficient transmission.

- The channelized uplink is unable to statistically multiplex bursty packet data generated by client-server interaction.

- The dynamically-assigned uplink is able to effectively statistically multiplex client-server data traffic offering minimal transmission delay and full resource utilization.

- For the same amount of uplink capacity, the dynamically-assigned uplink architecture is able to support more simultaneous sessions than the channelized uplink architecture.

The superiority of the dynamically-assigned uplink architecture is demonstrated by its ability to handle both message transfer from bursty sources and client-server traffic with minimum delay. Additionally, the added flexibility of adaptively assigning capacity as system loading fluctuates allows the dynamically-assigned uplink to achieve higher maximum utilization than a mutli-channel uplink.

We have seen that an unbiased evaluation of the future trends in satellite communications must be undertaken before any new satellite communication architectures are proposed. Also, the development of effective techniques for modeling new and emerging communication requirements is necessary in order to predict the performance of candidate architectures. Finally, with careful determination of pertinent performance metrics, we can determine the superiority of designs which meet today's communication needs as well as those of the future.

Chapter 3

General Packet Arrival Models

3.1 Overview

In this chapter we investigate packet arrival models beyond the traditional Poisson arrival model used in Chapter 1. The Poisson arrival process is generally used to model the arrivals both from a single source and, because of the superposition properties of Poisson point processes, composite arrivals from a large population. In each case we assume the individual source is a constant Poisson source, that is, the source is always "on" and producing packets with independent and exponentially distributed interarrival times.

With the recent explosion of network-based applications and services, packet source modeling has become increasingly important to facilitate the accurate assessment of the performance of proposed system architectures. Further understanding of packet arrival statistics and the development of effective modeling techniques is paramount to specifying architectures which will meet emerging user requirements.

We can consider many sources which we previously categorized as bursty to be, yet again, bursty. The traditional Poisson point process offers the bursty arrival characteristics of individual packets, but many sources also show burstiness in the mean arrival rate of packets. Such sources are said to be ***doubly-stochastic*** in the sense that there are two random processes affecting packet arrival statistics. The *Markov-modulated Poisson process* (MMPP) is an arrival process which can accurately model the doubly-stochastic nature of some sources. Specifically, the MMPP is a Poisson process with a mean arrival rate which varies according to a second, independent, Markov process. An example would be the packet arrivals from a user's client application as presented in Chapter 2. Here the source is in two states: an active state where the source is "on" and producing packets and an inactive state where the source is "off". MMPPs can be used to accurately model this type of source.

Results from the investigation of MMPPs lead to queuing models which assume MMPPs for the arrival process, such as the MMPP/G/1 queue. Exact analysis is possible, but with high computational complexity for models with many states. MMPPs with only two states, however, are found to be fairly accurate at approximating general sources generated with MMPPs of many states. In addition, MMPPs with only two states can be used to approximate correlated sources such as packetized voice and video. Efficient algorithms exist for solving two-state MMPPs offering an attractive technique for approximating these unique arrival processes.

Finally we consider literature discussing empirical measurements of network traffic, that is, aggregate packet arrivals from a large number of diverse sources. It is found that the commonly used methods for modeling individual sources do not necessarily scale when considering network traffic. In fact, network traffic is found to have large variation in arrival rate over all time scales and is termed *self-similar*. We briefly discuss self-similarity and a few of the techniques to model sources that exhibit the phenomenon. Further investigation of appropriate models of self-similar sources will allow accurate specification of communication resources able to handle the long-term variation of network traffic.

3.2 The Markov-Modulated Poisson Process

3.2.1 Definition

The Markov-modulated Poisson process (MMPP), fully described in [14], is a doubly stochastic Poisson process whose arrival rate is time-varying and given by

$$\lambda(t) = \lambda_{J(t)}, \qquad t \geq 0 \tag{3.1}$$

where $J(t)$ is an m-state irreducible Markov process and λ_i is the Poisson arrival rate when in state i. Specifically, we can construct an MMPP by varying the arrival rate of a Poisson process according to an irreducible, discrete-state, continuous-time, Markov process which is independent of the arrival process. The continuous-time Markov process with m states (assuming homogeneity) is described by the infinitesimal generator matrix

$$\mathbf{Q} = \begin{bmatrix} -\sigma_1 & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & -\sigma_2 & \cdots & \sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \sigma_{m2} & \cdots & -\sigma_m \end{bmatrix}. \tag{3.2}$$

The Poisson arrival rates for each of the m states are grouped together in the vector

$$\underline{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_m \end{bmatrix}^{\mathrm{T}} \tag{3.3}$$

and for later utility, in the diagonal form

$$\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m). \tag{3.4}$$

The steady-state vector of the arrival rate Markov process is $\underline{\pi}$ such that

$$\underline{\pi}\,\mathbf{Q} = 0 \tag{3.5}$$

and

$$\underline{\pi}\,\underline{e} = 1 \tag{3.6}$$

where $\underline{e}$ is the all-ones vector of length m, that is

$$\underline{e} = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^{\mathrm{T}}. \tag{3.7}$$

3.2.2 Poisson Process

As one would imagine, the traditional Poisson arrival process with mean arrival rate λ is a special case of an MMPP with $m = 1$ state and $\lambda_1 = \lambda$. Note that an MMPP with $m > 1$ and $\lambda_i = \lambda$ for all i is also the traditional Poisson arrival process with mean arrival rate λ.

3.2.3 MMPP(2) Specialization

The simplest non-degenerate case of the MMPP is the 2-state ($m = 2$) process denoted MMPP(2). The Markov process infinitesimal generator matrix for the MMPP(2) simplifies to

$$\mathbf{Q} = \begin{bmatrix} -\sigma_1 & \sigma_1 \\ \sigma_2 & -\sigma_2 \end{bmatrix} \tag{3.8}$$

while the arrival rate matrix follows as

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \tag{3.9}$$

The steady-state vector $\underline{\pi}$ of the Markov process for the MMPP(2) can be directly calculated from

$$\underline{\pi} = \begin{bmatrix} \pi_1 & \pi_2 \end{bmatrix} = \frac{1}{\sigma_1 + \sigma_2} \begin{bmatrix} \sigma_2 & \sigma_1 \end{bmatrix}. \tag{3.10}$$

3.2.4 Interrupted Poisson Process

The interrupted Poisson process (IPP) is a Poisson process which alternates between an "on" state with arrivals occurring with a mean rate λ and an "off" state where no arrivals occur. The process is either "on" or "off" for exponentially distributed lengths of time. The IPP is a special case of the MMPP(2) where the arrival rate of one state is zero such that

$$\Lambda = \begin{bmatrix} \lambda & 0 \\ 0 & 0 \end{bmatrix} \tag{3.11}$$

and the transitions between the "on" and "off" states are determined by the state transition rate matrix $\mathbf{Q}$.

3.2.5 Superposition of n Identical MMPP(2)s

To model a finite number of independent and identical arrival processes described by MMPP(2)s, we can superimpose each process to develop one large MMPP. Thus for an individual arrival process described by (3.8) and (3.9), the superposition of n users is described by an $m = n + 1$ state MMPP with infinitesimal generator

$$\begin{array}{ll} [\mathbf{Q}_n]_{i,i} = -i\sigma_1 - (n-i)\sigma_2, & \text{for } 0 \le i \le n \\ [\mathbf{Q}_n]_{i,i-1} = i\sigma_1, & \text{for } 1 \le i \le n \\ [\mathbf{Q}_n]_{i,i+1} = (n-i)\sigma_2, & \text{for } 0 \le i \le n-1 \\ [\mathbf{Q}_n] = 0, & \text{otherwise} \end{array} \tag{3.12}$$

and

$$\Lambda_n = \operatorname{diag}(i\lambda_1 + (n-i)\lambda_2, 0 \le i \le n). \tag{3.13}$$

3.2.6 MMPP/G/1 Queues

We can now consider queuing models with more intricate customer arrival processes described with the above MMPP framework. Thus for a general service time distribution, using queuing theory nomenclature, we have an MMPP/G/1 queue where the mean customer arrival rate is Poisson and time-varying according to $\mathbf{Q}$ and Λ. We define the mean arrival rate as

$$\lambda_{\text{tot}} = \underline{\pi}\, \underline{\lambda} \tag{3.14}$$

where $\underline{\pi}$ is the steady-state vector of $\mathbf{Q}$. As imagined, the M/G/1 queue is a special case of the general framework of the MMPP/G/1 queue that follows.

With the arrival process described by $\mathbf{Q}$ and Λ, we characterize the general service time with the distribution[1] $\widetilde{H}(x)$ with finite mean h and second and third non-central moments $h^{(2)}$ and $h^{(3)}$. Additionally, the Laplace-Stieltjes transform of $\widetilde{H}(x)$ is denoted as $H(s)$, that is,

$$H(s) = \int_0^\infty e^{-sx} d\widetilde{H}(x), \tag{3.15}$$

and will be used in solving the MMPP(2)/G/1 queue described below.

MMPP/G/1 Queuing Delay Determination

If we define $\mathbf{G}_{ij}$ as the probability that a busy period starting with the MMPP in state i ends in state j, then we can extract the embedded Markov process of the MMPP/G/1 queue. The matrix $\mathbf{G}$ is determined by solving

$$\mathbf{G} = \int_0^\infty e^{(\mathbf{Q}-\Lambda+\Lambda\mathbf{G})x} d\widetilde{H}(x) \tag{3.16}$$

[1] Note the notation of $\widetilde{H}(x)$ does indeed denote the (cumulative) ***distribution*** of the service time, ***not*** the density.

with the technique presented below. Noting $\mathbf{G}$ is a stochastic matrix, the steady-state vector $\underline{g}$ of $\mathbf{G}$ satisfies both

$$\underline{g}\,\mathbf{G} = \underline{g} \tag{3.17}$$

and

$$\underline{g}\,\underline{e} = 1 \tag{3.18}$$

as before.

Surprisingly it is only $\underline{g}$ which we need, in addition to the other definitions above, to calculate the moments of the queue wait time. The first moment of the MMPP/G/1 waiting time distribution is given by[14][15]

$$w_v = \frac{1}{2(1-\rho)}\left[2\rho + \lambda_{\text{tot}}h^{(2)} - 2h\left((1-\rho)\underline{g} + h\underline{\pi}\Lambda\right)\left(\mathbf{Q} + \underline{e}\,\underline{\pi}\right)^{-1}\underline{\pi}\right]. \tag{3.19}$$

Additionally, the second moment of waiting time distribution is given by

$$w_v^{(2)} = \frac{1}{3(1-\rho)}\left[3h\left(2\underline{W_1}(0)(h\Lambda - I) - h^{(2)}\underline{\pi}\Lambda\right)\left(\mathbf{Q} + \underline{e}\,\underline{\pi}\right)^{-1}\underline{\lambda} - 3h^{(2)}\underline{W_1}(0)\underline{\lambda} + \lambda_{\text{tot}}h^{(3)}\right] \tag{3.20}$$

where

$$\underline{W_1}(0) = \left(h\underline{\pi}\,\Lambda + (1-\rho)\underline{g}\right)\left(\mathbf{Q} + \underline{e}\,\underline{\pi}\right)^{-1} - \underline{\pi}(1 + w_v). \tag{3.21}$$

MMPP/G/1 Algorithm

As mentioned above, it is the steady-state vector $\underline{g}$ of the embedded Markov process described by $\mathbf{G}$ that is needed to solve the MMPP/G/1 queue wait delay statistics. Thus we need an efficient method for solving the non-linear matrix integral equation of (3.16) for $\mathbf{G}$. Efficient solutions to this equation are presented in [16][17][18][19][14] for the general m-state MMPP with the most recent results described with the following algorithm:

Initial Step: Define

$$\mathbf{G}_0 = 0 \tag{3.22}$$

$$\mathbf{H}_{0,k} = I, \qquad k = 0, 1, 2, \ldots \tag{3.23}$$

$$\Theta = \max_i\left((\Lambda - \mathbf{Q})_{ii}\right) \tag{3.24}$$

$$\gamma_n = \int_0^\infty e^{-\Theta x}\frac{(\Theta x)^n}{n!}\mathrm{d}\tilde{H}(x), \qquad n = 0, 1, \ldots, n^* \tag{3.25}$$

where n^* is chosen such that

$$\sum_{k=0}^{n^*}\gamma_k > 1 - \varepsilon_1, \qquad \varepsilon_1 \ll 1. \tag{3.26}$$

Recursion: For $k = 0, 1, 2, \ldots$ compute

$$\mathbf{H}_{n+1,k} = \left[I + \frac{1}{\Theta}\left(\mathbf{Q} - \Lambda + \Lambda\mathbf{G}_k\right)\right]\mathbf{H}_{n,k}, \qquad n = 0, 1, \ldots, n^* \tag{3.27}$$

$$\mathbf{G}_{k+1} = \sum_{n=0}^{n^*}\gamma_n\mathbf{H}_{n,k}. \tag{3.28}$$

Stopping Condition: End the above recursion when

$$\|\mathbf{G}_{k+1} - \mathbf{G}_k\| < \varepsilon_2, \qquad \varepsilon_2 \ll 1 \tag{3.29}$$

and declare the solution to be $\mathbf{G} = \mathbf{G}_{k+1}$.

MMPP(2)/G/1 Algorithm

As imagined, the computation required to solve (3.16) is greatly simplified for the $m = 2$ MMPP(2)/G/1 queue[16]. For a given $\mathbf{Q}$, Λ, and $H(s)$, we calculate our solution $\mathbf{G}$ by the following algorithm:

Initial Step: Define

$$G_1 = 0. \tag{3.30}$$

Recursion: Compute cyclically

$$G_2 = \frac{G_1\sigma_1}{\sigma_1 + G_1(\lambda_1 - \lambda_2)} \tag{3.31}$$

$$G_1 = 1 - G_2 - H(\sigma_1 + \sigma_2 + \lambda_1 G_1 + \lambda_2 G_2). \tag{3.32}$$

Stopping Condition: Continue the above cyclical computation until G_1 and G_2 become stable. If, by chance, G_1 does not converge and takes on negative values, then exchange indices and restart algorithm. Declare the solution to be $\mathbf{G}$ as per

$$\mathbf{G} = \begin{bmatrix} 1 - G_1 & G_1 \\ G_2 & 1 - G_2 \end{bmatrix}. \tag{3.33}$$

For the MMPP(2) we can solve for the steady-state vector $\underline{g}$ with

$$\underline{g} = \begin{bmatrix} g_1 & g_2 \end{bmatrix} = \frac{1}{G_1 + G_2} \begin{bmatrix} G_2 & G_1 \end{bmatrix}. \tag{3.34}$$

3.2.7 Applications of MMPPs

MMPP(2) Approximations to General Arrival Processes

We can use the MMPP(2) to approximate an arrival processes by matching the first three moments of the arrival rate $\lambda(t)$ of the process with the moments of the MMPP(2)[20][21][22][23]. Doing so allows us to use the computational reduction of the MMPP(2)/G/1 queue algorithm to approximate the behavior of a queue with a general arrival process. The arrival process to be approximated may either be obtained from empirical measurements of an actual arrival process or some reduction of an analytic MMPP source with many states.

Queue Overflow Models

MMPPs play a role in modeling the overflows from finite-capacity trunks[22][24]. In [22] the overflow arrivals are modeled as an IPP allowing an exact analysis of the steady-state behavior of the MMPP/M/c/($c+k$) queue with c servers. The investigation in [24] considers the analysis of the MMPP/G/1/k single-server queue.

Packetized Voice and Data Traffic Multiplexing

The problem of modeling the superposition of packetized voice traffic and data traffic onto a single channel is investigated in [20] and [21]. The voice traffic consists of fixed-length packets arriving with a constant rate (i.e. the arrivals are highly correlated) while the data traffic is bursty in nature. Techniques of determining the queuing delay associated with such a system are developed with an MMPP(2)/G/1 queue to assess the QoS guarantees for the delay-sensitive voice traffic.

Queuing Theory

The MMPP is shown to be a special case of the N/G/1 queue in [25] where the N-process is a general point process introduced by Neuts[26][18]. Additionally the MMPP is shown to be a special case of the ***batch Markovian arrival process*** in [16]. Single-server queues with vacations are investigated in [17] for a class of non-renewal arrival processes of which the MMPP is a special case. Finally, additional investigations are performed in [27] and [28] to develop faster (although more complex) algorithms to solve the general non-linear matrix equation of (3.16).

3.3 Self-Similar Traffic

3.3.1 Overview

The traffic models investigated so far are appropriate for modeling message arrivals from a single user or an aggregation of multiple users. For relatively short time scales both models capture the burstiness of arrivals from individual users. Additionally, both the Poisson point process and the Markov-modulated Poisson point process predict some constant mean arrival rate while observing the process over a large time duration. That is, these models predict a *smoothing effect* of message arrivals both as the number of aggregate users increases and as the arrival process is viewed over longer periods of time.

Actual observation of Ethernet local area network (LAN) traffic, however, shows quite the opposite situation. While investigating LAN traffic (packet arrivals from many diverse sources) over long time intervals we continue to see large fluctuation of arrival rate, that is, the assumption of traffic smoothing is *not* observed. Additionally, as the number of bursty sources increases, from observation, the aggregate traffic source typically exhibits *increased* burstiness[29]. Thus the Poisson and Poisson-related models are appropriate for short-term traffic modeling, but these models do not accurately capture the long-term variability of aggregated traffic sources.

Network traffic is shown in many investigation to be statistically *self-similar*, that is, we observe burstiness on all time scales. It is this large-scale variation that is not captured in the Poisson traffic models. We briefly discuss self-similarity below and techniques for modeling self-similar sources.

3.3.2 Recent Results

An ambitious effort was made in [30] to record the packet traffic on an active Ethernet LAN in a research facility. Hundreds of millions of packets were observed with their time of arrival recorded with high accuracy. Sample results of this recording process are shown in plots (a) through (e) of Fig. 3.1. We see the expected bursty nature for plots (d) and (e) for small time frames less than 1 [sec]. When considering plots (a) through (c) for larger time frames, however, we still see large-scale variation of arrival rate. For comparison, plots (a') through (e') show sample paths generated from an appropriately-chosen compound Poisson traffic model. Note how the results from the Poisson model do not accurately represent reality as we compare packet arrival statistics over larger periods of time.

3.3.3 Modeling Self-Similar Sources

Variability on all time scales is characteristic of fractal-like behavior formulated in chaos theory. Mandelbrot coined the term "self-similar" to describe processes which appear scale-invariant, that is, the statistics of large sample intervals are similar to those for small sample intervals. The self-similarity of LAN traffic is quite different from the statistics of conventional telephone traffic and from the Poisson-related stochastic models commonly used in their analysis.

To model self-similar traffic it is important to characterize the burstiness of the traffic source. Methods of measuring burstiness are presented in [30] and [31] using an ***index of dispersion*** metric to capture characteristics of the source beyond second-order statistics. Once burstiness

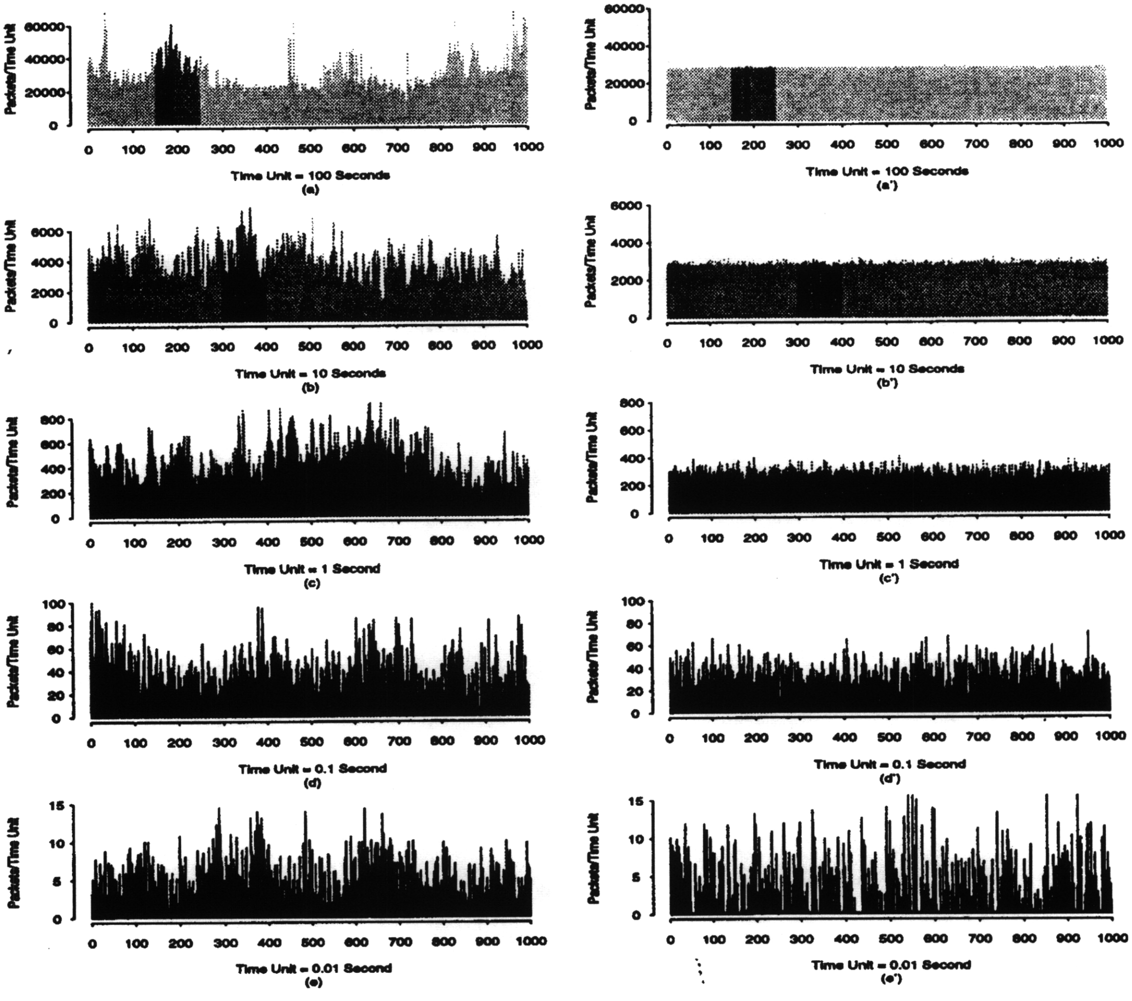


Figure 3.1: Measured Ethernet traffic presented as packets per time unit for (a) 100 [sec], (b) 10 [sec], (c) 1 [sec], (d) 0.1 [sec], and (e) 0.01 [sec]. For comparison purposes, synthetic traffic from a Poisson arrival model is shown for the same five time scales in (a') through (b').[30]

is quantified, we can use the MMPP(2)[2] to model a self-similar traffic source with measured dispersion[31][32]. Another alternative to modeling a given self-similar source is to model the data transfer process (both short-term protocol interaction and long-term usage statistics) for a given application (such as VBR video traffic) and show the resulting self-similar nature of the generated traffic[33].

3.3.4 Summary

The above account briefly describes investigations into characterizing traffic statistics from multiple diverse packet sources. By identifying network traffic as being statistically self-similar we are able to develop traffic models which resemble long-term packet arrival statistics of actual networks. Adequate models of network traffic allow us to properly assess the performance of network trunking, such as satellite extensions to ground-based networks.

[2] Note here we are using the MMPP(2) to simplify the computational burden of modeling the overall self-similar arrival process. We are not using the MMPP(2) to model individual sources.

Chapter 4

Generalized Multiple Access and Multiple Access Coding

4.1 Overview

We consider multiple access in a more general form than first introduced in Chapter 1 by abstracting the problem into vector space. To begin, we develop frequency, time, and code division multiplexing as a vector space problem and show the equivalency of these three multiplexing techniques. With the vector space view of multiplexing we are then able to generalize orthogonal multiple access techniques to develop a geometrical description of the problem of allowing multiple users to share a common communication resource.

The abstraction of multiple access to vector space also holds while considering random access techniques where contention for resources exist. Slotted ALOHA, as discussed in Chapter 1, is shown to be a specific case of the framework developed in this chapter.

Continuing with the vector space development of random access we consider a slotted ALOHA transmission technique that allows the simultaneous decoding of more than one packet transmission per slot. This technique, referred to as m-S-ALOHA, builds on the code division multiplexing development, but with the variation of non-orthogonal codes. Thus we adapt our vector space abstraction of multiple access to include techniques of describing multiple access interference. Methods of comparing m-S-ALOHA to traditional S-ALOHA are presented, taking into consideration the bandwidth expansion of multiple access coding.

4.2 Generalization of Multiple Access to Vector Space

4.2.1 Orthogonal Multiplexing

The orthogonal multiplexing techniques of TDM, FDM, and CDM were introduced in Chapter 1 to begin our discussion of multiple access communications. We now present a more geometrical description of these three orthogonal multiplexing techniques.

Time Division Multiplexing

The idea behind time division multiplexing (TDM) is to divide access time to a communication channel between multiple services. Consider N independent services wishing to utilize a communication channel within a time interval of duration T. Each service is allowed one contiguous amount of access time to the channel. If we allow the i^{th} service to access the channel for a duration of T_i, then it must be

$$\sum_{i=1}^{N} T_i = T. \tag{4.1}$$

To avoid interference from other services we mandate exclusive channel access to the i^{th} service during its assigned access interval of duration T_i.

The above formulation of time division multiplexing smacks of orthogonality. Let us define the function

$$\psi_i(t) = p_{T_i}\left(t - \sum_{j=1}^{i-1} T_i\right) \tag{4.2}$$

where $p_T(t)$ is the unit pulse defined by

$$p_T(t) = \begin{cases} 1, & 0 \leq t < T \\ 0, & \text{otherwise} \end{cases}. \tag{4.3}$$

Additionally from this formulation we see the inner product

$$<\psi_i(t), \psi_j(t)> = \int \psi_i(t)\psi_j^*(t)dt = 0 \qquad \text{for } i \neq j, \qquad 1 \leq i, j \leq N. \tag{4.4}$$

Thus we have abstracted TDM to vector space by defining a ***basis vector*** $\psi_i(t)$ for the i^{th} service to use to access the channel. The orthogonality condition of (4.4) ensures there is no interference between services. Hence we have a complete orthogonal set $\{\psi_i(t)\}_{i=1}^{N}$ of basis functions in N-dimensional space. For each time interval of T the i^{th} service may access the channel during the support of its basis function $\psi_i(t)$.

Frequency Division Multiplexing

The same vector space abstraction suggested for TDM can also be applied to frequency division multiplexing (FDM). In this case we have a bandwidth of W which must be shared between multiple services. With a contiguous band of frequencies of width W_i assigned to the i^{th} service we have, similar to above,

$$\sum_{i=1}^{N} W_i = W. \tag{4.5}$$

The basis vectors for FDM are taken as

$$\psi_i(f) = p_{W_i}\left(f - \sum_{j=1}^{i-1} W_i\right) \tag{4.6}$$

where $p_W(f)$ is the frequency domain analogue of $p_T(t)$, viz.,

$$p_W(f) = \begin{cases} 1, & 0 \leq f < W \\ 0, & \text{otherwise} \end{cases}. \tag{4.7}$$

We have the same orthogonality condition for each service with this formulation of FDM when

$$<\psi_i(f), \psi_j(f)> = \int \psi_i(f)\psi_j^*(f)df = 0 \qquad \text{for } i \neq j, \qquad 1 \leq i, j \leq N, \tag{4.8}$$

resulting in a complete orthogonal set of basis functions $\{\psi_i(f)\}_{i=1}^{N}$ spanning N-space. Here the i^{th} service may signal only in the support of $\psi_i(f)$ in the frequency domain.

Code Division Multiplexing

We allow a more general form for the basis functions of code division multiplexing (CDM). For both TDM and FDM above, we made the implicit assumption that the region of support of the respective basis function was a connected region. For CDM, however, we allow the region of support of each basis function to be composed of multiple individually-connected regions. Thus, in the time domain, we define a basis vector for the i^{th} service to be of the form

$$\psi_i(t) = \sum_{j=1}^{M} c_{ij} p_{T_c}(t-(j-1)T_c) \tag{4.9}$$

with each $c_{ij} \in \{0,1\}$ and $p_T(t)$ defined, as before, with (4.3).

For CDM the basic pulse shape $p_T(t)$ is termed a *chip* with *chipping rate* of $\frac{1}{T_c}$. The sequence $\{c_{ij}\}_{j=1}^{M}$ is referred to as the *code* for the i^{th} service. If we define the vector

$$\underline{c}_i = \left[\begin{array}{cccc} c_{i1} & c_{i2} & \cdots & c_{iM} \end{array}\right] \tag{4.10}$$

then orthogonality between services is assured when

$$<\underline{c}_i, \underline{c}_j> = \underline{c}_i \underline{c}_j^{\mathrm{H}} = \sum_{k=1}^{M} c_{ik} c_{jk} = 0 \qquad \text{for } i \neq j, \qquad 1 \leq i,j \leq N. \tag{4.11}$$

The i^{th} service is offered access to the channel for a duration of T_i for each time interval of T. Furthermore, we must assure each service has access to an integer number of chip intervals, hence we must obey

$$\sum_{i=1}^{N} r_i T_c = M T_c = T \tag{4.12}$$

where r_i is the number of chip intervals within the interval T assigned to the i^{th} service.

To allow N services to coexist without interfering with each other it is necessary to have N orthogonal basis functions $\{\psi_i(t)\}_{i=1}^{N}$. The orthogonality between basis functions implies we must have N orthogonal codes $\{\underline{c}_i\}_{i=1}N$, further implying the constraint on code length of

$$M \geq N. \tag{4.13}$$

Note for $M = N$ and $T_c = \frac{T}{N}$ we have the same formulation for TDM as described above when all users have equal channel access intervals, that is, when $T_i = \frac{T}{N}$ for all i. We can consider TDM to be a special case of this more general CDM formulation.

Equivalency of Multiplexing Techniques

It is important to note that the above three multiplexing techniques are all equivalent mathematically. That is, they all utilize the communication resources fully and offer the same overall transmission capacity to each service. To show this we consider a transmission scheme that offers a bandwidth efficiency of B [bits/sec/Hz] and calculate the total transmission capacity of the three multiplexing techniques.

$$\begin{aligned} C_{\mathrm{TDM}} &= \sum_{i=1}^{N} \int \psi_i(t) B \, dt \\ &= \sum_{i=1}^{N} \int p_{T_i}\left(t - \sum_{j=1}^{i-1} T_i\right) B \, dt \\ &= \sum_{i=1}^{N} T_i B \\ &= TB \text{ [bits/Hz]} \end{aligned} \tag{4.14}$$

$$
\begin{aligned}
C_{\mathrm{FDM}} &= \sum_{i=1}^{N} \int \psi_i(f) B \; df \\
&= \sum_{i=1}^{N} \int p_{W_i} \left(f - \sum_{j=1}^{i-1} W_i \right) B \; df \\
&= \sum_{i=1}^{N} W_i B \\
&= WB \text{ [bits/sec]}
\end{aligned}
\tag{4.15}
$$

$$
\begin{aligned}
C_{\mathrm{CDM}} &= \sum_{i=1}^{N} \int \psi_i(t) B \; dt \\
&= \sum_{i=1}^{N} \int \sum_{j=1}^{M} c_{ij} p_{T_c} \left(t - (j-1)T_c \right) B \; dt \\
&= \sum_{i=1}^{N} r_i T_c B \\
&= M T_c B \\
&= TB \text{ [bits/Hz]}
\end{aligned}
\tag{4.16}
$$

Normalizing by either unit-time ($T = 1$ [sec]) or unit-bandwidth ($W = 1$ [Hz]) we see TDM, FDM, and CDM all offer the same overall transmission capacity.

4.2.2 Orthogonal Multiple Access

As described in Chapter 1, multiple access is the process of assigning communication resources between a group of users. From our formulation above we see this process reduces to the assignment of available basis functions to the users (what we termed as *services* above) needing to share the communication resource. Again, because all basis functions of the multiplexing techniques above are orthogonal, all users sharing the communication resource do so without interfering with each other.

4.2.3 Contention-mode Multiple Access

We can consider random access within the scope of vector space when considering slotted transmission techniques. Here we assume a set of basis functions $\{\psi_i(\cdot)\}_{i=1}^{N}$ are available from one of the three multiplexing techniques described above. We further assume a large population of users and hence cannot assign a single basis function to each user. During each slot, users with new packet arrivals must *contend* for one of the N basis functions and, if two or more users choose the same basis function for a given slot, a collision occurs. If all transmitting users happen to each choose a unique basis function $\psi_i(\cdot)$ then a maximum of N packets may be sent in the duration of one slot.

We can consider S-ALOHA to be the special case of the $N = 1$ single-dimension vector space where users contend for a single basis function, i.e. the shared channel. We can also consider a system offering N independent channels with users randomly choosing a channel for transmission. In this case each channel corresponds to an FDM basis function. Regardless of the multiplexing technique, we are able to abstract the problem to a vector space formulation.

4.3 Multiple Access Coding

In the last section we learned how to visualize random access communication in vector space. With N available basis functions, each transmitting user randomly chooses a basis function for transmission during a given slot. As mentioned, if two or more users happen to pick the same basis function a mutual collision occurs. We term this to be a *hard collision* as both transmissions completely interfere with each other.

We can, however, decide to allow a transmitting user form some *linear combination* of the basis functions to be used for transmission during the slot interval. In this case, the mutual interference which we term *multiple access interference* two users would experience would simply be the projection of one user's linear combination onto the other. For small interference we say the users have experienced a *soft collision* which may or may not inhibit proper decoding of the transmission. If we consider the weights of the linear combination to be a *code*, we now have the problem of minimizing the mutual interference between two or more transmissions by selecting appropriate codes, i.e. the problem of *multiple access coding.*

4.3.1 Vector Space Formulation

To begin the development of this concept, let us define the basis functions of our vector space to be

$$\phi_i(t) = p_{T_c}\left(t - (i-1)T_c\right) \tag{4.17}$$

so that we form the orthogonal set of basis functions $\{\phi_i(t)\}_{i=1}^{N}$. Notice each individual basis function is a shifted version of the *chip* we introduced above with duration T_c. For the duration of the slot, user i sends with on-off signaling using the signal

$$s_i(t) = \sum_{j=1}^{N} c_{ij}\phi_j(t) \tag{4.18}$$

with $c_{ij} \in \{0, 1\}$. As before, if we define the vector

$$\underline{c}_i = \left[\begin{array}{cccc} c_{i1} & c_{i2} & \cdots & c_{iN} \end{array}\right] \tag{4.19}$$

we say $\underline{c}_i$ is the *code* which user i modulates to convey each bit during the slot with on-off signaling. Thus, during the entire slot, the i^{th} user will send

$$\begin{aligned} w_i(t) &= \sum_{k=0}^{L-1} b_i s_i(t - kNT_c) \\ &= \sum_{k=0}^{L-1}\sum_{j=1}^{N} b_i c_{ij}\phi_j\left(t - kNT_c - (i-1)T_c\right)) \end{aligned} \tag{4.20}$$

with $\{b_i\}_{i=1}^{L}$ being the bit sequence to be conveyed.

4.3.2 Multiple Access Interference

Now that we have formulated the waveform each transmitting user will use to send its information, we now must quantify the multiple access interference that may occur between multiple user transmissions in a given slot. Clearly, since the i^{th} user signals each bit with the same code $\underline{c}_i$ for the entire slot duration, we only need to consider multiple access interference for the transmission of a single bit.

Ideally, if the number of users wishing to send is less than N we can assign the i^{th} user to use the code

$$c_{ij} = \begin{cases} 1, & j = i \\ 0, & \text{otherwise} \end{cases} \tag{4.21}$$

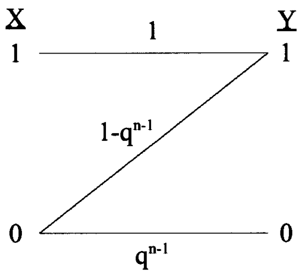


Figure 4.1: The Z channel model for the multiple access interference channel.

and we can be assured of no interference between users. Unfortunately the number of users is greater than N and we cannot guarantee orthogonality. In this case we use

$$\sigma_{ij} = <s_i(t), s_j(t)> = \underline{c}_i \underline{c}_j^{\mathrm{T}} \tag{4.22}$$

to measure the interference between user i transmitting with code $\underline{c}_i$ and user j transmitting with code $\underline{c}_j$. We can now visualize multiple access interference in vector space as simply the inner product between transmitted codes. Note also $\sigma_{ij} = \sigma_{ji}$, that is, the amount user i interferes with j is the same amount user j interferes with user i.

4.3.3 Coding

The problem of multiple access coding is now formulated as finding a codebook $\mathcal{C}_m$ consisting of m N-length sequences $\underline{c}_i$ which minimize multiple access interference, that is,

$$\mathcal{C}_m = \left\{ \{\underline{c}_i\}_{i=1}^{m} | \min_{\underline{c}_i, \underline{c}_j} \sum_{i=1}^{m} \sum_{j=i+1}^{m} \underline{c}_i \underline{c}_j^{\mathrm{H}} \right\}. \tag{4.23}$$

With a codebook $\mathcal{C}_m$ we are assured of having the set of codes which minimize multiple access interference for a system which can decode up to m simultaneous transmissions per slot.

4.3.4 Capacity Reduction due to Multiple Access Interference

Our transmission technique above assumes an on-off channel where, for any bit interval, the channel is assumed to be "on" if one or more users transmit or "off" if no users transmit. We can determine the capacity of this channel by assuming the only source of noise is multiple access interference[34].

Let us assume we have a total of n identical users. We define q as the probability a given user will *not* transmit during a given bit interval and note the channel formed is the classical Z channel shown in Fig. 4.1 with crossover probability

$$\rho = 1 - q^{n-1}. \tag{4.24}$$

The capacity of this instance of the Z channel is given by

$$C = \max_{q} \left[H_b\left(q^n\right) - q H_b\left(q^{n-1}\right) \right] \tag{4.25}$$

where $H_b(p)$ denotes the binary entropy function

$$H_b(p) = -p \log(p) - (1-p)\log(1-p). \tag{4.26}$$

Since we are interested in large user populations, the duty cycle of each user must be small and hence q must be close to unity. If we parameterize q with k by defining

$$q = 1 - \frac{k}{n}, \tag{4.27}$$

we have, for large n,

$$\begin{aligned} q^n &= \left(1-\frac{k}{n}\right)^n \\ &= 1-n\frac{k}{n}+\frac{n(n-1)}{2!}\left(\frac{k}{n}\right)^2-\frac{n(n-1)(n-2)}{3!}\left(\frac{k}{n}\right)^3+\cdots \\ &\approx 1-k+\frac{1}{2!}k-\frac{1}{3!}k^3+\cdots \\ &= e^{-k}. \end{aligned} \tag{4.28}$$

Our capacity calculation of (4.25) becomes

$$\begin{aligned} C &\approx \max_k \left[H_b\left(e^{-k}\right)-\left(1-\frac{k}{n}\right)H_b\left(\frac{e^{-k}}{1-\frac{k}{n}}\right)\right] \\ &= \max_k \left[-\frac{k}{n}\log\left(1-e^{-k}\right)\right] \\ &= \frac{\ln 2}{n}. \end{aligned} \tag{4.29}$$

For TDMA a user would be assigned $\frac{1}{n}$ of the available capacity. Therefore we conclude that multiple access interference reduces achievable channel capacity by a factor of $\ln 2 \approx 0.695$ relative to TDMA.

4.4 Analysis of Coded Slotted ALOHA Systems

We have now introduced a few concepts of multiple accessing in vector space under the label of multiple access coding. Continuing in this direction, we discuss the combination of multiple access coding and traditional S-ALOHA in this section.

Recalling from Chapter 1, with traditional S-ALOHA only one packet transmission is allowed per time slot. If two or more users simultaneously transmit during a given slot, all transmission will be undecodable and we declare a mutual collision. With multiple access coding, however, we can develop an ALOHA technique which allows for more than one successful packet transmission per time slot. We call this transmission technique m-S-ALOHA to differentiate it from traditional S-ALOHA.

What follows is a general development of m-S-ALOHA presented in [11]. We assume some arbitrary underlying transmission scheme and packet arrival model while developing the steady-state throughput of m-S-ALOHA. Continuing, we assume Poisson arrivals and show, for $m = 1$, this formulation generalizes to that of traditional S-ALOHA. Finally, after discussing the importance of bandwidth normalization, we discuss two candidate multiple access coding techniques and compare their delay versus throughput with traditional S-ALOHA.

4.4.1 Packet Success Probability

The foundation of generalizing S-ALOHA to m-S-ALOHA is the ability to quantify the probability of successful decoding of m simultaneous packet transmissions during any given time slot. It is assumed that the underlying transmission scheme (currently unspecified) can offer $p_E(m)$, the probability a particular packet will *not* be decodable given the fact that m simultaneous transmissions were attempted in the given time slot. From this we define

$$p_C(m) = 1 - p_E(m) \tag{4.30}$$

as the packet success probability and further assume the success of a given packet is independent of the outcomes of all prior time slots. We further assume all m packet attempts during a given time slot experience the same probability of decoding error $p_E(m)$.

As with traditional S-ALOHA, we assume the dominant source of interference is that of other contending users, what we termed above as multiple access interference. We therefore recognize, assuming no background noise,

$$p_C(1) = 1 \tag{4.31}$$

for all transmission schemes to be considered.

As stated, $p_C(m)$ will be determined by the underlying transmission technique. We note, however, for the analysis of traditional S-ALOHA we assume the packet success probability to be

$$p_C(m) = \begin{cases} 1, & m = 1 \\ 0, & \text{otherwise} \end{cases}, \tag{4.32}$$

that is, any single packet attempt is always successful, but multiple packet attempts per time slot always result in a collision.

It is also possible to consider a threshold approximation to the packet success probability as in [35]. We assume a packet transmission is successful if only K_{max} maximum simultaneous transmissions are attempted. In this case we see

$$p_C(m) = \begin{cases} 1, & 1 \leq m \leq K_{max} \\ 0, & \text{otherwise} \end{cases}. \tag{4.33}$$

4.4.2 Steady-State Throughput

With the definition of the packet success probability $p_C(m)$, we are now able to develop the expected number of successful packet transmissions per slot. For a given time slot with M packet transmissions attempted, the probability of K successful packet transmissions is given by

$$\Pr[K = k|M = m] = \binom{m}{k} p_C^k(m) p_E^{m-k}(m). \tag{4.34}$$

In order to determine the steady-state throughput we must first characterize the composite packet arrival process. In place of immediately assuming Poisson arrivals, we develop our analysis in a more general case. We define $f_M(m)$ as the probability of there being exactly m packet transmission attempts during any given slot and develop popular packet arrival processes below.

By assuming the $M - K$ unsuccessful packet attempts are transmitted after some random delay, we can determine the steady-state throughput to be the expected number of successful packet transmission per slot. Specifically, let us define S as the steady-state throughput of the m-S-ALOHA channel where S is measured in packet transmissions per time slot. We see

$$\begin{aligned} S &= \mathrm{E}\{K\} \\ &= \mathrm{E}\{\mathrm{E}\{K|M\}\} \\ &= \mathrm{E}\left\{\sum_{k=0}^{M} k \binom{M}{k} p_C^k(M) p_E^{M-k}(M)\right\} \\ &= \mathrm{E}\{M p_C(M)\} \\ &= \sum_{m=1}^{\infty} m p_C(m) f_M(m) \end{aligned} \tag{4.35}$$

where (4.35) follows for our characterization of the composite arrival process.

4.4.3 Composite Packet Arrival Models

As stated above, our development has been independent of the packet arrival process injecting packets into the m-S-ALOHA channel. Although, in general, any desired arrival process can by specified, it is the binomial and Poisson models which are most commonly used.

If we assume our user population to be finite with N users then the number of users M attempting to transmit a packet in the given slot is a random variable distributed according to the probability mass function

$$f_M(m) = \begin{cases} \binom{N}{m} p^m (1-p)^{N-m}, & 0 \le m \le N \\ 0, & \text{otherwise} \end{cases} \tag{4.36}$$

where p is the probability a given user will transmit a packet in the given time slot.

The Poisson model follows from the binomial model as N becomes large and, consequently, p becomes small. Thus, assuming an infinite population, the number of packet attempts M in the given slot is characterized by

$$f_M(m) = \begin{cases} \frac{(\lambda T)^m}{m!} e^{-\lambda T}, & 0 \le m \\ 0, & \text{otherwise} \end{cases} \tag{4.37}$$

where T is the slot duration in [sec]. As experienced previously, the parameter λ is interpreted as the composite arrival rate of packets from *all* users, measured in [packets/sec].

4.4.4 Poisson Arrival Specialization

For small user populations the binomial packet arrival model gives exact system performance evaluation. For large populations, as described above, the Poisson model is quite adequate at approximating the exact binomial model and offers reduced computational complexity. We therefore adopt the Poisson model and integrate it into our formulation of m-S-ALOHA.

To begin, we define the offered traffic load to the m-S-ALOHA channel, measured in packets per time slot, as

$$G = \lambda T \tag{4.38}$$

where, again, λ is the composite packet arrival rate from the whole population of users. Next, we substitute (4.37) into (4.35) with

$$\begin{aligned} S &= \sum_{m=1}^{\infty} m p_C(m) f_M(m) \\ &= \sum_{m=1}^{\infty} m p_C(m) \frac{G^m}{m!} e^{-G} \\ &= G e^{-G} \sum_{m=0}^{\infty} \frac{G^m}{m!} p_C(m+1) \\ &= G e^{-G} \left\{ p_C(1) + G p_C(2) + \frac{G^2}{2!} p_C(3) + \frac{G^3}{3!} p_C(4) + \cdots \right\} \\ &= G e^{-G} \left\{ 1 + G p_C(2) + \frac{G^2}{2!} p_C(3) + \frac{G^3}{3!} p_C(4) + \cdots \right\}. \end{aligned} \tag{4.39}$$

With throughput expanded as in (4.39) we see the additional packet transmission ability offered by supporting multiple packets per slot.

We can consider an idealized system where the underlying transmission scheme can support an infinite number of packet transmissions per time slot and assure successful decoding. In this case, we have $p_C(m) = 1$ for all m allowing (4.39) to become

$$\begin{aligned} S_{\mathrm{PC}} &= Ge^{-G}\left\{1 + G + \frac{G^2}{2!} + \frac{G^3}{3!} + \cdots\right\} \\ &= Ge^{-G}\left\{e^G\right\} \\ &= G \end{aligned} \tag{4.40}$$

and the m-S-ALOHA channel can handle all traffic offered to it. We can consider this limiting case be the that of "perfect coding", that is, the underlying multiple access coding scheme is able to perfectly separate user transmissions and avoid interference between users.

As a second limiting case, we can consider the packet success probability for traditional S-ALOHA given by (4.32), that is, we allow only one packet transmission per time slot. Substituting (4.32) into (4.39) yields

$$S_{\mathrm{S-ALOHA}} = Ge^{-G} \tag{4.41}$$

which has become quite familiar from our treatment of S-ALOHA in Chapter 1.

From these two special cases we conclude that any proposed m-S-ALOHA transmission scheme will exhibit a steady-state throughput $S_{m-\mathrm{S-ALOHA}}$ obeying

$$S_{\mathrm{S-ALOHA}} < S_{m-\mathrm{S-ALOHA}} < S_{\mathrm{PC}}. \tag{4.42}$$

We are, however, ignoring bandwidth in the above comparison. Clearly the "perfect-coding" scenario would require bandwidth in great excess of that required for traditional S-ALOHA, with m-S-ALOHA falling somewhere in between. In order to fairly compare the throughput of m-S-ALOHA we must consider the bandwidth consumed by the transmission scheme.

4.4.5 Bandwidth Normalization

We define the *bandwidth expansion factor* B as the ratio of operating bandwidth of the proposed m-S-ALOHA scheme to the bandwidth of traditional S-ALOHA, that is,

$$B = \frac{W_{m-\mathrm{S-ALOHA}}}{W_{\mathrm{S-ALOHA}}}. \tag{4.43}$$

Obviously, for traditional S-ALOHA we have $B = 1$ and for any transmission scheme which allows more than one successful packet transmission per time slot will have $B > 1$.

We consider comparing the steady-state throughput of m-S-ALOHA by defining *utilization* $\hat{\mathcal{S}}$ as being the bandwidth-normalized throughput, viz.,

$$\hat{\mathcal{S}} = \frac{S}{B}. \tag{4.44}$$

Now, with utilization (steady-state throughput per unit bandwidth) $\hat{\mathcal{S}}$, we have a fair method of comparing the performance of various m-S-ALOHA techniques with each other and with traditional S-ALOHA.

4.4.6 Example Transmission Techniques

The above development of m-S-ALOHA has assumed the existence of some transmission technique able to support multiple successful packet transmissions per time slot. Furthermore, we have assumed the transmission scheme may be completely characterized by the packet success probability $p_C(m)$ and the associated bandwidth expansion factor B.

We now specialize by summarizing two representative transmission schemes analyzed in [11]. The first technique is *time hopping multiple access* (THMA), described in [36] and [37], and the second is *pulse position tree coded multiple access* (TCMA), described in [34]. Briefly, THMA operates by replacing each message bit by a Q-length binary sequence with k ones, operating

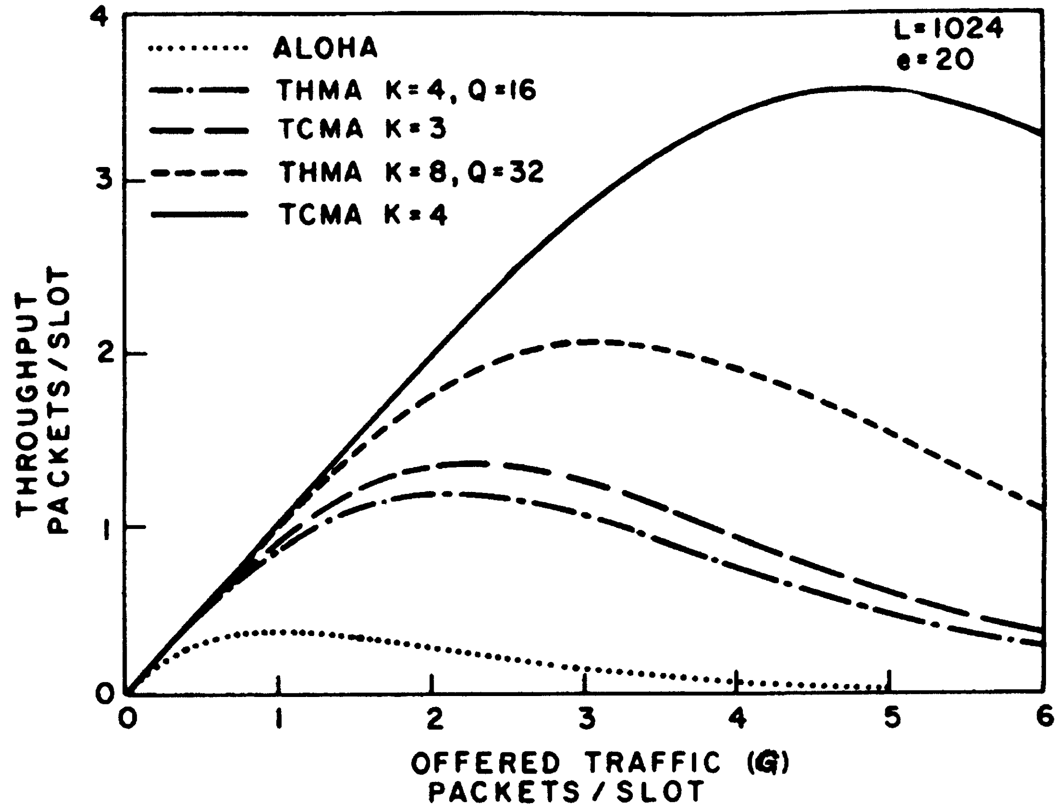


Figure 4.2: Throughput S versus offered traffic G for traditional S-ALOHA and m-S-ALOHA using THMA and TCMA.[11]

with a bandwidth expansion factor of $B_{\text{THMA}} = Q$. With TCMA, on the other hand, the message is encoded using a convolutional tree code of constraint length K, with output symbols being one of 2^K orthogonal pulses for each message bit. Thus the bandwidth expansion for TCMA is $B_{\text{TCMA}} = 2^K$.

Both THMA and TCMA can be visualized in the vector space development presented above. For THMA operating with Q-length binary sequences with k ones, our vector space is of dimension Q and each packet transmission uses a code which is a linear combination of k basis functions. For TCMA with constraint length K, we are in a vector space of dimension 2^K with each packet transmission using a code composed of only one basis function.

The throughput performance of the two transmission schemes is shown in Fig. 4.2. Notice how the throughput versus offered load is constrained between the traditional S-ALOHA performance and that of perfect coding. We see, as the complexity of the multiple access coding increases (i.e. as Q and/or K are increased), we are able to support more packet transmissions per time slot. We have not, however, considered the increased bandwidth consumption of each transmission scheme.

In Fig. 4.3 we see expected delay versus utilization shown for S-ALOHA, THMA m-S-ALOHA, and TCMA m-S-ALOHA. Note the abscissa is our defined utilization $\hat{S}$ and does indeed consider the bandwidth expansion of each transmission technique. Although Fig. 4.2 demonstrated the superior performance of THMA and TCMA over traditional S-ALOHA, this figure does not consider the additional bandwidth consumed by THMA and TCMA. We see the true picture in Fig. 4.3, however, and, with proper bandwidth normalization, THMA actually performs *poorer* than traditional S-ALOHA. For most cases of TCMA we, again, have poorer

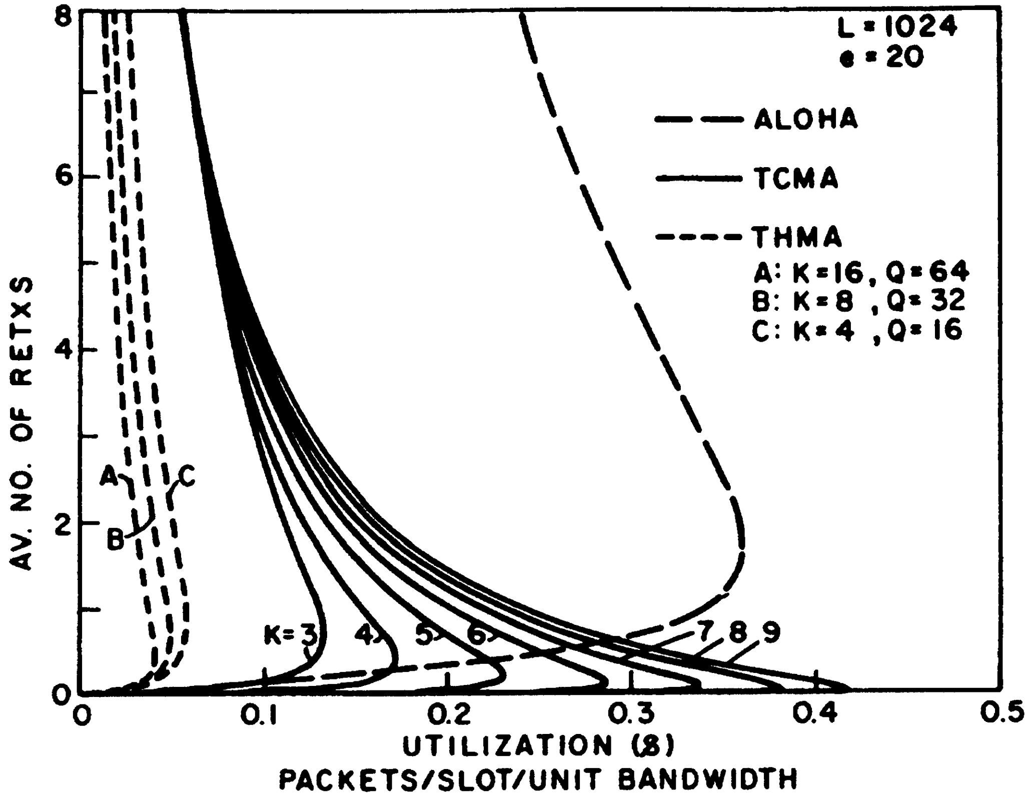


Figure 4.3: Delay versus utilization $\hat{S}$ (bandwidth-normalized throughput) characteristics for traditional S-ALOHA and m-S-ALOHA using THMA and TCMA. Note the inferior performance of THMA compared with S-ALOHA and the superior performance of TCMA for large constraint lengths K.[11]

performance than that achieved with traditional S-ALOHA. However, when we can afford the large bandwidth expansion of TCMA with large constraint length K, TCMA offers *additional* steady-state transmission throughput beyond that offered by traditional S-ALOHA. Note also the distinctively different delay versus utilization characteristic of TCMA m-S-ALOHA. Minimal delay is offered to packet arrivals throughout all utilization values below the maximum achievable utilization.

4.4.7 Other Sources of Information

There is much literature investigating various aspects of combining multiple access coding with ALOHA transmission. We have summarized the results in [11] to develop a general framework to consider various underlying transmission schemes while also introducing the performance of two example transmission techniques. What follows is a brief summary of other pertinent literature.

In addition to THMA (discussed in [36] and [37]) and TCMA (discussed in [34]), multiple access coded transmission techniques are also discussed in [38] and [39]. This latter reference considers very low rate convolutional codes for both multiple access interference and background Gaussian noise mitigation.

Upper and lower bounds to the packet success probability $p_C(m)$ are investigated in [35] and [40] assuming random signature sequences. Both of these references investigate the generally-assumed Gaussian approximation to the probability of bit error while [35] also considers correlated bit errors within a single packet transmission. In [41] bounds on packet success probability are developed when the worst-case, chip-synchronous interference assumption is lifted.

Consideration of incorporating block coding with each packet transmission in order to overcome multiple access interference is investigated in [11] and [35]. In [11] it is shown that combining BCH block coding with each packet transmission increases overall throughput while considering the increase in bandwidth necessary to allow block coding. Numerical results are presented in [35] while the rate of block coding is varied.

Non-ideal factors such as timing errors and fading channels are also covered. Packet time of arrival errors are investigated in [42] and [43] where it is found that multiple access coding decreases the necessary guard time between packet transmissions. In [44] it is shown multiple access coded m-S-ALOHA benefits from using maximal ratio combining of the path diversity inherent with multipath transmission environments, a performance increase unobtainable with traditional uncoded S-ALOHA.

The combination of multiple access coding and S-ALOHA is further developed in [45] while considering the ***single-hop network*** topology. A general framework is developed which allows for the investigation of effects of code assignment, channel coding, capture effects, and scheduling disciplines.

4.4.8 Summary

We have generalized multiplexing and ALOHA transmission in this chapter by considering these topics from a vector space perspective. Multiple access communications is now viewed as the process of assigning vector space basis functions to services wishing to transmit information. This formulation allows us to view the vector space inner product as a measure of the multiple access interference between users and gives us a metric for developing effective multiple access coding schemes.

The generalization of Slotted ALOHA developed in this chapter allows us to consider advanced forms of random access, namely the inclusion of coding that both separates users transmission as much as possible, but also aids in mitigating the remaining multiple access interference and background noise. With random access coding we now consider media access control and channel signaling to be a combined optimization problem.

Chapter 5

Conclusion

We have been introduced to the topic of multiple access transmission techniques for satellite communication systems. Notions of orthogonal multiple access and random access transmission have been developed with a discussion which summarizes known transmission techniques. In addition, performance metrics of various multiple access schemes have been identified.

Our abstraction of multiplexing and multiple accessing to vector space offers us an intuitive visual framework for conceptualizing various transmission techniques. Both orthogonal multiple access transmission and random access (or contention-mode) transmission fit into this vector space formulation as either the assignment of or contention for the orthogonal basis functions of the space. With this development we introduced the notion of multiple access coding. With coding the multiple access problem is formulated as how best to situate users in a high dimensional space such that we minimize the multiple access interference between simultaneous transmissions. As an example of multiple access coding we discussed a generalization of traditional Slotted ALOHA transmission called m-S-ALOHA. By relying on some underlying transmission technique to minimize interference, we are able to allow multiple successful transmissions per time slot and hence greater throughput than traditional Slotted ALOHA.

The Poisson arrival process has served communication engineering well for many decades with its ability to model the arrival of telephone calls, messages, and, more recently, packets from data communication services and networks. As data communication services evolve, however, it has been found that the Poisson arrival process is not versatile enough to model real-world packet arrival processes. Our introduction to the Markov-modulated Poisson process (MMPP) generalizes the traditional Poisson model and offers a mathematical framework to affect more exacting performance evaluation of emerging communication services. Additionally, we have discussed the utility of the MMPP for approximating complex packet arrival processes, such as those experienced with local area network data traffic which has been shown to be statistically self-similar. Although more computationally complex, today's computing resources allow for the higher fidelity offered by these generalized arrival processes for the modeling of both packet sources and the networks which offer packet transport.

Finally, our investigation of two different satellite uplink architectures offers an unbiased evaluation of several multiple access transmission techniques for handling bursty data sources. The channelized uplink architecture is shown to be inherently non-optimal for transmitting packet data due to the fractional transmission capacity assigned to any given user. The dynamically-assigned uplink architecture offers overall transmission delay which is far superior to that obtainable with the channelized uplink architecture. The ability of the dynamically-assigned uplink architecture to benefit from a reconfigurable uplink offers additional performance gain unobtainable with a fixed channelized uplink.

In addition to considering both the channelized and dynamically-assigned uplink architectures, attention was given to assess the communication requirements of future users of satellite communication systems. Unscheduled message transfer must undoubtedly be offered by any future satellite communication system, but more importantly, it is becoming necessary to offer network access via satellite. In this direction, we have investigated both uplink architectures to assess their performance for efficient transmission of data from user client applications to the server which holds the desired information. Again, the dynamically-assigned uplink architecture shows higher performance over the channelized uplink both in regards to overall message delay, but also in its ability to efficiently statistically multiplex multiple concurrent client-server sessions.

It is the tools and techniques presented above with which future satellite communication system architectures should be designed and evaluated. Continual assessment of future communication requirements is essential for developing systems which will efficiently offer communication resources to tomorrow's satellite communication users. Forgoing past methodologies and system designs allows the broad consideration of many emerging communication techniques and system architectures. With careful evaluation of all available options coupled with an understanding of data communication trends, we can confidently move forward with satellite communication systems that will continue to be extremely effective at meeting user demands and requirements far beyond one generation of satellite hardware.

Appendix A

Key Results of Queuing Theory

A.1 Summary of Queuing Models

A.1.1 Overview

As noticed by the contents of this Thesis, performance analysis of message transfer is primarily investigated through the framework of queuing theory. We have used various queuing models to develop expected delay while investigating multiple access transmission techniques and candidate satellite uplink architectures. What follows is a brief summary of the functional queuing delay models $\mathcal{Q}_{\mathrm{M/M/k}}(\cdot)$ and $\mathcal{Q}_{\mathrm{M/G/k}}(\cdot)$ used throughout the preceding analysis. The reader interested in developing a firm understanding of queuing theory should consult [13] and [7].

To differentiate various delay models, queuing theory has adopted the notation of $\mathcal{A}/\mathcal{B}/\mathcal{C}$ to describe the nature of system arrivals and service disciplines, as follows:

Parameter $\mathcal{A}$ Indicates the nature of the customer arrival process where we generally find M for *memoryless* interarrival processes (such as the Poisson arrival process), G for *general* interarrival processes, or D for *deterministic* interarrival processes.

Parameter $\mathcal{B}$ Indicates the nature of the service process, that is, how customers leave the queuing model. The same elements M, G, or D as above are used to characterize the service time distribution as being either exponential, general, or deterministic, respectively.

Parameter $\mathcal{C}$ Indicates the number of servers available with the queuing model.

We focus on memoryless arrival processes due to our adoption of the Poisson process for user message arrivals. All three service time distributions are also covered since we assume message lengths to be either fixed (deterministic service) or exponentially distributed (exponential service). Note deterministic service is a simple special case of general service. Finally, we cover single server systems as well as systems with multiple servers.

A.1.2 M/M/k Queues

M/M/k Solution

Queuing models associated with memoryless arrivals of mean arrival rate λ and exponentially distributed service times of mean $\frac{1}{\mu}$ behave as follows. The *utilization factor* ρ $(0 \leq \rho < 1)$ for a k-server system is defined as

$$\rho = \frac{\lambda}{k\mu}. \tag{A.1}$$

The probability that the system will be idle is given by

$$p_0 = \left[\sum_{n=0}^{k-1} \frac{(k\rho)^n}{n!} + \frac{(k\rho)^k}{k!(1-\rho)}\right]^{-1} \tag{A.2}$$

allowing the determination of the probability that there are one or more arrivals in the system waiting for service

$$P_Q = \frac{p_0(k\rho)^k}{k!(1-\rho)}. \tag{A.3}$$

The expected delay an arrival experiences once entering the system is

$$\mathcal{Q}_{\mathrm{M/M/k}}(\lambda, \mu, k) = \frac{\rho P_Q}{\lambda(1-\rho)} \tag{A.4}$$

and once one of the k servers becomes available, the expected service time is

$$T = \frac{1}{\mu}. \tag{A.5}$$

Thus an arrival to the system experiences an expected delay of

$$W = \frac{\rho P_Q}{\lambda(1-\rho)} + \frac{1}{\mu} \tag{A.6}$$

before service is complete. Additionally, note the utilization factor ρ, for a $k > 1$ server system, is the normalized number of busy servers. The total number of customers within the system (both waiting for service and receiving service) is given by

$$N = k\rho + \frac{\rho P_Q}{1-\rho} \tag{A.7}$$

M/M/1 Specialization

For a single server system we have the following:

$$\begin{aligned}
\rho &= \frac{\lambda}{\mu} && \text{(A.8)}\\
\mathcal{Q}_{\mathrm{M/M/1}}(\lambda, \mu) &= \frac{\rho}{\mu - \lambda} && \text{(A.9)}\\
T &= \frac{1}{\mu} && \text{(A.10)}\\
W &= \frac{\rho}{\mu - \lambda} && \text{(A.11)}\\
N &= \frac{\lambda}{\mu - \lambda} && \text{(A.12)}\\
& && \text{(A.13)}
\end{aligned}$$

M/M/∞ Specialization

A particularly interesting case is when we allow the number of servers k to tend toward infinity. In this situation we assume a new server is created to immediately handle a new arrival. Thus the expected delay an arrival experiences is zero (exceptional service) and the queue behaves as:

$$\begin{aligned}
\mathcal{Q}_{\mathrm{M/M/\infty}} &= 0 && \text{(A.14)}\\
T &= \frac{1}{\mu} && \text{(A.15)}\\
W &= \frac{1}{\mu} && \text{(A.16)}\\
N &= \frac{\lambda}{\mu} && \text{(A.17)}\\
& && \text{(A.18)}
\end{aligned}$$

A.1.3 M/G/k Queues

M/G/1 Solution

We characterize general service distributions with the first and second moments of the service time:

$$\begin{aligned}\overline{X} &= \mathrm{E}\left\{X\right\} = \text{Expected service time} && \text{(A.19)}\\ \overline{X^2} &= \mathrm{E}\left\{X^2\right\} = \text{Second moment of service time} && \text{(A.20)}\end{aligned}$$

For the single server system ($k = 1$) we can obtain the exact solution for first-order queue statistics:

$$\begin{aligned}\rho &= \lambda\overline{X} && \text{(A.21)}\\ \mathcal{Q}_{\mathrm{M/G/1}}\left(\lambda,\overline{X},\overline{X^2}\right) &= \frac{\lambda\overline{X^2}}{2(1-\rho)} && \text{(A.22)}\\ T &= \overline{X} && \text{(A.23)}\\ W &= \frac{\lambda\overline{X^2}}{2(1-\rho)}+\overline{X} && \text{(A.24)}\\ N &= \rho+\frac{\lambda^2\overline{X^2}}{2(1-\rho)} && \text{(A.25)}\end{aligned}$$

M/G/k Approximation

For the general k-server system we can approximate the first-order queue statistics from first and second-order service statistics[46]

$$\begin{aligned}\overline{X} &= \mathrm{E}\left\{X\right\} = \text{Expected service time} && \text{(A.26)}\\ \overline{X^2} &= \mathrm{E}\left\{X^2\right\} = \text{Second moment of service time.} && \text{(A.27)}\end{aligned}$$

Note what follows is indeed an exact solution for the k-server system if the service times are exponentially distributed with mean $\overline{X} = \frac{1}{\mu}$ and second moment $\overline{X^2} = \frac{2}{\mu^2}$.

$$\begin{aligned}\rho &= \frac{\lambda\overline{X}}{k} && \text{(A.28)}\\ \mathcal{Q}_{\mathrm{M/G/k}}\left(\lambda,\overline{X},\overline{X^2},k\right) &\approx \frac{\lambda^k\overline{X^2}\left(\overline{X}\right)^{k-1}}{2(k-1)!\left(k-\lambda\overline{X}\right)^2\left(\sum_{n=0}^{k-1}\frac{\left(\lambda\overline{X}\right)^n}{n!}+\frac{\left(\lambda\overline{X}\right)^k}{(k-1)!\left(k-\lambda\overline{X}\right)}\right)} && \text{(A.29)}\\ T &= \overline{X} && \text{(A.30)}\\ W &= \mathcal{Q}_{\mathrm{M/G/k}}(\cdot)+\overline{X} && \text{(A.31)}\end{aligned}$$

A.1.4 M/D/k Queues

As alluded to above, deterministic (fixed-length) service times with mean $\frac{1}{\mu}$ are handled with the general service time distribution results with moments

$$\begin{aligned}\overline{X} &= \frac{1}{\mu} && \text{(A.32)}\\ \overline{X^2} &= \frac{1}{\mu^2}. && \text{(A.33)}\end{aligned}$$

That is to say,

$$\begin{aligned}\mathcal{Q}_{\mathrm{M/D/1}}\left(\lambda,\mu\right) &= \mathcal{Q}_{\mathrm{M/G/1}}\left(\lambda,\frac{1}{\mu},\frac{1}{\mu^2}\right) && \text{(A.34)}\\ \mathcal{Q}_{\mathrm{M/D/k}}\left(\lambda,\mu,k\right) &= \mathcal{Q}_{\mathrm{M/Gk}}\left(\lambda,\frac{1}{\mu},\frac{1}{\mu^2},k\right). && \text{(A.35)}\end{aligned}$$

A.2 Multiple Channel Systems

A.2.1 Problem Statement

Consider the problem of two interconnected communication nodes where Node 1 receives messages which must be sent to Node 2 with minimum delay. Connecting Node 1 to Node 2 are N identical communication channels with capacity of $\frac{C}{N}$ [bits/sec] each. Node 1 receives messages addressed to Node 2 according to a Poisson arrival process with mean arrival rate λ [msgs/sec]. The lengths of the messages are exponentially distributed with a mean length of $\frac{1}{\mu}$ [bits]. When a new message arrives at Node 1 it is serviced by any available channel connecting to Node 2. If, however, all channels connecting Node 1 to Node 2 are occupied, the new arrival waits at Node 1 and is serviced in a first-come first-served manner.

Our interest, as with all message transfer performance analysis, is to minimize the expected delay a message experiences while being transmitted from Node 1 to Node 2. The overall communication capacity between Node 1 and Node 2 is C [bits/sec], but how many channels N should this capacity be divided into to minimize delay?

A.2.2 Analysis

To answer this question we first formulate the expected delay T_N a single message experiences while being transmitted from Node 1 to Node 2 using one of N channels, as developed in [47]. This delay is composed of both the expected delay waiting for one of the N channels to become available and, once service begins, the actual transmission delay over the channel of capacity $\frac{C}{N}$ [bits/sec]. Thus we wish to choose N to minimize

$$\begin{aligned} T_N &= \mathcal{Q}_{\mathrm{M/M/k}}\left(\lambda, \frac{\mu C}{N}, N\right) + \frac{N}{\mu C} \\ &= \frac{N}{\mu C}\left[1 + \frac{1/N(1-\rho)}{S_N(1-\rho)+1}\right] \qquad \text{(A.36)} \end{aligned}$$

where

$$\rho = \frac{\lambda}{\mu C} \qquad \text{(A.37)}$$

and

$$S_N = \sum_{n=0}^{N-1} \frac{(N\rho)^{n-N} N!}{n!}. \qquad \text{(A.38)}$$

Investigating S_N we discover

$$\begin{aligned} S_N &= \sum_{n=0}^{N-1} \rho^{n-N} \frac{N^{n-N} N!}{n!} \qquad \text{(A.39)} \\ &= \sum_{n=0}^{N-1} \rho^{n-N} \frac{N}{N}\frac{N-1}{N} \cdots \frac{n+1}{N} \\ &\leq \sum_{n=0}^{N-1} \rho^{n-N} \qquad \text{(A.40)} \\ &= \frac{\rho^{-N}-1}{1-\rho} \qquad \text{(A.41)} \end{aligned}$$

allowing us to obtain the bound

$$0 < S_N \leq \frac{\rho^{-N}-1}{1-\rho} \qquad \text{for N} \geq 1. \qquad \text{(A.42)}$$

Applying this bound on S_N to (A.36) yields

$$\begin{aligned} T_N &\geq \frac{N}{\mu C}\left[1+\frac{\rho^N}{N(1-\rho)}\right] && \text{(A.43)} \\ &= \frac{N(1-\rho)+\rho^N}{\mu C(1-\rho)} && \text{(A.44)} \end{aligned}$$

and with

$$\alpha = 1-\rho \qquad \text{(A.45)}$$

we see that for $0<\rho<1$

$$\begin{aligned} N(1-\rho)+\rho^N &= N\alpha+(1-\alpha)^N && \text{(A.46)} \\ &\geq N\alpha+1-N\alpha && \text{(A.47)} \\ &= 1. && \text{(A.48)} \end{aligned}$$

Thus for all $N \geq 1$ we have the lower bound on delay

$$T_N \geq \frac{1}{\mu C(1-\rho)} \qquad \text{(A.49)}$$

which, incidentally, reduces to

$$T_N \geq T_1. \qquad \text{(A.50)}$$

Therefore, to minimize the expected message transmission delay from Node 1 to Node 2, we should have all available transmission capacity C assigned to only one channel, viz.,

$$\underset{N\geq 1}{\operatorname{argmin}}\, T_N = 1. \qquad \text{(A.51)}$$

Any division of C into separate channels will result in greater message delay and will be non-optimal.

Appendix B

Capacity Assignment in Split-Channel Reservation Systems

B.1 Overview

We have introduced the notion of reservation access systems in Chapter 1 and have investigated their performance in Chapter 2. An important question which arises when considering these reservation systems is the optimal division of uplink capacity between the reservation channel and data channel. Optimality, in this sense, is referred to as the capacity assignment which minimizes expected message transmission delay. Total uplink capacity must be balanced between the reservation process and message transmission process so that overall delay is minimized. Of additional interest is the concept of *reconfigurable communication architectures* where the capacity assignment between reservations and message transfer can dynamically adjust to changing user requirements and system loading.

Optimal capacity assignment is considered for the split-channel reservation system shown in Fig. B.1. Here we divide overall uplink capacity between reservations and message transfer with the parameter θ. Thus, for an uplink with total capacity C [bits/sec], the reservation channel receives $(1-\theta)C$ [bits/sec] of capacity while the remaining θC [bits/sec] of capacity handle message transfer. The value of θ, for given system parameters, which minimizes the overall expected message transmission delay is declared as optimum.

The optimal capacity assignment is first investigated for an idealized perfect scheduling model. As in previous chapters, perfect scheduling assumes we have a "genie-aided" system where all users have instantaneous feedback and automatic coordination of the data channel is assumed. Next we consider the inclusion of multiple access on the reservation channel. Numerical results are presented for the optimal capacity assignment when reservations are placed with the

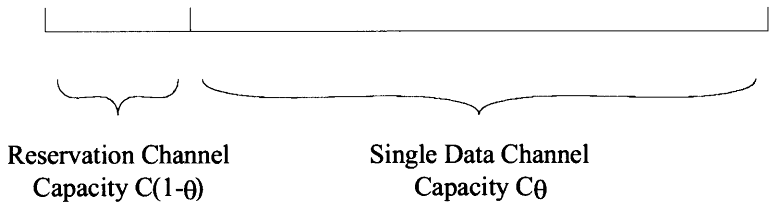


Figure B.1: The split-channel reservation system assigns $1-\theta$ % of available uplink capacity to reservations while the remaining capacity is devoted to the data channel.

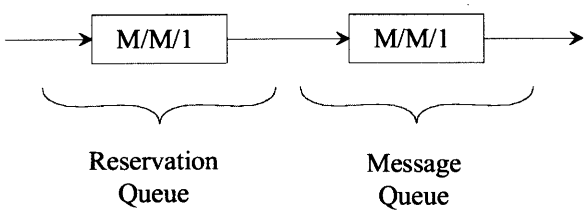


Figure B.2: The perfectly-scheduled split-channel reservation system can be analyzed by assuming two M/M/1 queues in series.

Slotted ALOHA random access transmission technique.

B.2 Capacity Assignment for Perfect Scheduling

We first consider the optimal delay-minimizing capacity assignment, θ, for a reservation algorithm operating under the perfect scheduling assumption[1]. The delay associated with the overall message transmission can be viewed as the cascade of two queues, one for messages and one for reservations, as shown in Fig. B.2.

Taking the liberty of assuming the reservation packet length to be exponentially distributed allows us to apply *Burke's Theorem* to decompose the combined queues into two independent queues[13]. More specifically, by assuming new message arrivals to be Poisson with parameter λ and the reservation packet length to be exponentially distributed with length l_r [bits], the packet departure process of the reservation queue is Poisson with parameter λ. For convenience we also assume the message packet length to be exponentially distributed with mean l_m [bits]. Now, for a total communication capacity of C [bits/sec], the expected delay to transmit a message from source to destination is given by

$$\mathcal{W}_{\mathrm{PS}}(\theta) = \mathcal{Q}_{\mathrm{M/M/1}}\left(\lambda, \frac{(1-\theta)C}{l_r}\right) + \frac{l_r}{(1-\theta)C} + \mathcal{Q}_{\mathrm{M/M/1}}\left(\lambda, \frac{\theta C}{l_m}\right) + \frac{l_m}{\theta C} + T_R. \tag{B.1}$$

Note the functional dependence of (B.1) on θ, that is, as expected, adjustment of the capacity assignment θ between the reservation channel and data channel changes the expected message transmission delay. The optimal assignment of channel capacity is the value of θ which *minimizes* expected message transmission delay, i.e.,

$$\theta_{\mathrm{opt,PS}} = \operatorname*{argmin}_{\theta\in(0,1)} \mathcal{W}_{\mathrm{PS}}(\theta). \tag{B.2}$$

To develop a closed-form expression for $\theta_{\mathrm{opt,PS}}$ to solve (B.2) we expand (B.1) to

$$\mathcal{W}_{\mathrm{PS}}(\theta) = \frac{\theta C l_r + (1-\theta)C l_m - 2 l_r l_m \lambda}{(\theta C - \lambda l_m)\left((1-\theta)C - \lambda l_r\right)}. \tag{B.3}$$

Differentiating and selecting the solution offering a valid θ yields the desired result of optimal capacity assignment of

$$\theta_{\mathrm{opt,PS}} = \left(C - l_r\lambda + \lambda\sqrt{l_r l_m}\right)\frac{\sqrt{l_m}}{C\left(\sqrt{l_r}+\sqrt{l_m}\right)}. \tag{B.4}$$

An important insight offered by (B.4) is that the optimal capacity assignment is a function of λ. Thus as the message arrival rate fluctuates, an uplink architecture that allows for dynamic

[1] Note we are assuming we still have need for a reservation channel. In a true perfectly-scheduled system *all* capacity would be assigned to the data channel and none to the reservation channel since reservations would no longer be necessary.

capacity assignment adjustment according to

$$\theta_{\mathrm{opt,PS}} = \frac{\sqrt{l_m}}{\sqrt{l_r}+\sqrt{l_m}} + \lambda\left[\frac{l_m\sqrt{l_r} - l_r\sqrt{l_m}}{C\left(\sqrt{l_r}+\sqrt{l_m}\right)}\right] \tag{B.5}$$

will allow the highest instantaneous message transmission performance.

B.3 Capacity Assignment for S-ALOHA

With optimal capacity assignment investigated above for perfect scheduling, we now focus on including multiple access into our reservation access system[48]. Specifically we are interested in investigating the optimal capacity assignment while implementing the random access technique of Slotted ALOHA (S-ALOHA) on the reservation channel. Again, the optimal capacity assignment $\theta_{\mathrm{opt,S-ALOHA}}$ is the value of θ which minimizes the expected message transmission delay. The distinction between $\theta_{\mathrm{opt,PS}}$ from above and $\theta_{\mathrm{opt,S-ALOHA}}$ is that now we must offer additional capacity to the reservation channel to overcome the delay attributed to reservation collision resolution.

As above, we begin by developing the expected transmission delay to send a message from source to destination. Since we are interested in investigating S-ALOHA we assume fixed-length reservation packets of length l_r [bits]. Thus for a total uplink capacity of C [bits/sec], the delay associated with placing a reservation is

$$\begin{aligned} w_r(\theta) &= \mathcal{T}_{\mathrm{S-ALOHA}}\left(\lambda, \frac{l_r}{(1-\theta)C}\right) \\ &\approx \frac{3}{2}\frac{l_r}{(1-\theta)C} + \left[\exp\left(G(\theta)\right) - 1\right]\left[T_R + \frac{(K+2)l_r}{2(1-\theta)C}\right] + T_R. \end{aligned} \tag{B.6}$$

Note we must determine the S-ALOHA channel utilization $G(\theta)$, as described in Chapter 1, by solving

$$\frac{\lambda l_r}{(1-\theta)C} = G(\theta)\exp\left(-G(\theta)\right) \tag{B.7}$$

for $G(\theta) < 1$.

Message lengths are also assumed to be a fixed-length of l_m [bits]. Once the reservation is received the expected delay to send the message is given by

$$\begin{aligned} w_m(\theta) &= \mathcal{Q}_{\mathrm{M/D/1}}\left(\lambda, \frac{\theta C}{l_m}\right) + \frac{l_m}{\theta C} + T_R \\ &= \frac{l_m}{\theta C}\left[\frac{\frac{\lambda l_m}{\theta C}}{2\left(1 - \frac{\lambda l_m}{\theta C}\right)} + 1\right] + T_R. \end{aligned} \tag{B.8}$$

Thus the overall expected message transmission delay, including both reservation establishment and message transfer, is found by combining (B.6) and (B.8), that is,

$$\mathcal{W}_{\mathrm{S-ALOHA}}(\theta) = w_r(\theta) + w_m(\theta). \tag{B.9}$$

We plot the overall expected message delay $\mathcal{W}_{\mathrm{S-ALOHA}}(\theta)$ in Fig. B.3 for the 1 [Mbit/sec] uplink with $\lambda \in \{10, 50, 90\}$ [msgs/sec]. We see, for small arrival rates (ρ small), variation of θ produces minimal improvement in message transfer delay. For large arrival rates (ρ large), however, choosing θ carefully is exceedingly important.

Note, in general, we are violating Burke's Theorem since the departure process from the S-ALOHA reservation channel is *not* Poisson and service times are *not* exponentially distributed.

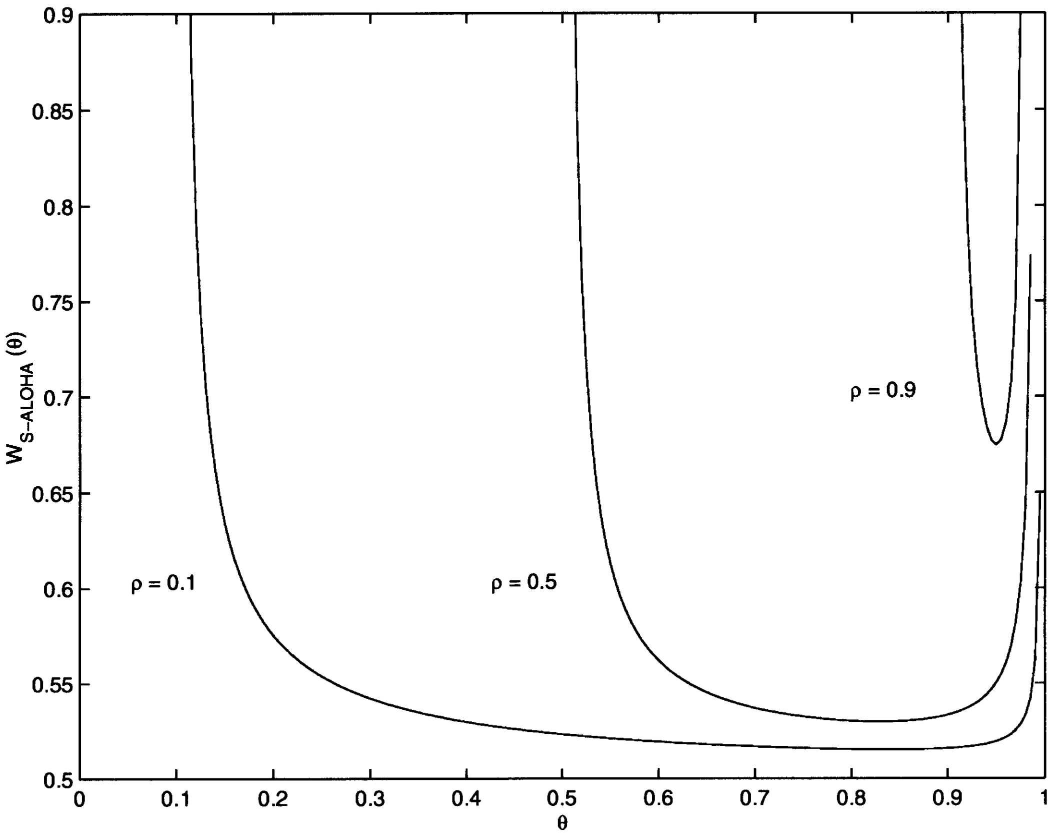


Figure B.3: The overall expected message delay $\mathcal{W}_{\text{S-ALOHA}}(\theta)$ is shown for all θ for the $C = 1$ [Mbit/sec] uplink. Note the increasing importance of optimizing θ as utilization ρ is increased.

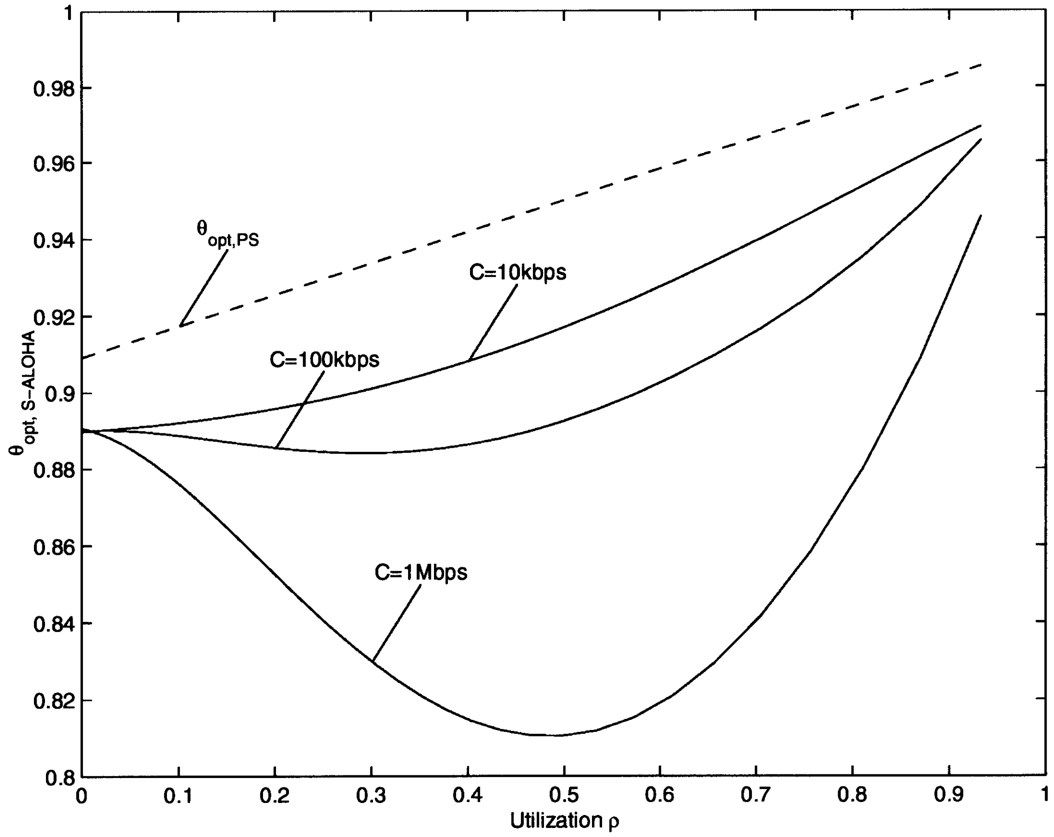


Figure B.4: Summary of numerical results for optimal capacity assignment $\theta_{\text{opt,S-ALOHA}}$ for S-ALOHA on reservation channel.

This decomposition is reasonably accurate, however, when we consider the fact that the departure process of the S-ALOHA reservation channel is sporadic (as collisions are resolved) and channel utilization is kept low (to maintain stability)[2].

As before, we wish to minimize $\mathcal{W}_{\text{S-ALOHA}}(\theta)$ by selecting θ according to

$$\theta_{\text{opt,S-ALOHA}} = \operatorname*{argmin}_{\theta \in (0,1)} \mathcal{W}_{\text{S-ALOHA}}(\theta). \tag{B.10}$$

Investigation of solving (B.10) in the light of (B.6) and (B.8) is surely a formidable task. We therefore resort to numerical techniques to solve (B.10) as a function of λ for values of l_r, l_m, and C of particular interest. Fig. B.4 summarizes the optimal capacity assignment $\theta_{\text{opt,S-ALOHA}}$ for reservation packet length $l_r = 100$ [bits], message packet length $l_m = 10$ [kbits], and overall uplink capacities of $C \in \{10, 100, 1000\}$ [kbits/sec]. The abscissa of Fig. B.4 is presented as message utilization ρ of capacity C [bits/sec]. Thus the actual message arrival rate λ varies for each curve from 0 [msgs/sec] to $\frac{C}{l_m} = \frac{C}{10^4}$ [msgs/sec]. In addition, for comparison purposes, we show $\theta_{\text{opt,PS}}$ from (B.5) for perfect scheduling in Fig. B.4.

It is not surprising to see $\theta_{\text{opt,PS}}$ in Fig. B.4 is greater than $\theta_{\text{opt,S-ALOHA}}$ since with perfect scheduling we have no contention on the reservation channel. To overcome the delay associated with collision resolution, the S-ALOHA reservation channel must be assigned additional capacity and thus we have

$$\theta_{\text{opt,S-ALOHA}} \leq \theta_{\text{opt,PS}}. \tag{B.11}$$

[2] A simulation of the Poisson departure process assumption is presented in [48] for a CSMA reservation channel. For low reservation channel utilization the assumption is reasonable.

From Fig. B.4 we see $\theta_{\text{opt,S-ALOHA}}$ to show more variation with utilization ρ as uplink capacity C increases. This is primarily because, as C increases, the mean message arrival rate λ is also increasing and hence more collisions are occurring on the reservation channel. To overcome the additional delay due to reservation collisions, more capacity is assigned to the reservation channel. As utilization increases further, however, the delay associated with the M/D/1 queue of the data channel becomes more significant and capacity assignment shifts back towards the data channel.

We can generalize our optimization by defining the parameters

$$\alpha = \frac{l_m}{l_r} \tag{B.12}$$

to capture the ratio of message length to reservation length (in bits) and

$$\beta = \frac{l_r}{C} \tag{B.13}$$

to hold the reservation packet transmission time assuming the whole uplink is available. Optimization continues as before, but now we are only concerned with the ratios α and β, which completely specify the operational characteristics of the S-ALOHA reservation channel and the data channel.

Fig. B.5 shows the optimal capacity assignment $\theta_{\text{opt,S-ALOHA}}$ for values of α and β of practical interest. Specifically, the following parameter values are shown:

α	β	Fig. B.5
10^2	10^{-4}	(1)
10^2	10^{-3}	(2)
10^2	10^{-2}	(3)
10^3	10^{-4}	(4)
10^3	10^{-3}	(5)
10^3	10^{-2}	(6)
10^4	10^{-4}	(7)
10^4	10^{-3}	(8)
10^4	10^{-2}	(9)

We note the cases with $\alpha = 10^2$ correspond to the previously investigated cases of $l_r = 10^2$ and $l_m = 10^4$ shown in Fig. B.4.

As expected, as we increase the ratio of message length to reservation packet length, greater capacity must be assigned to the data channel and hence $\theta_{\text{opt,S-ALOHA}}$ increases with α. Also, as α increases, we see capacity assignment is less dependent upon the ratio β. In this instance, overall transmission delay is highly dependent on the assignment of adequate resources to the data channel to handle these large message lengths.

In summary, we find exacting capacity allocation for the split-channel reservation system to be necessary to support high system loading with minimal delay. Arbitrary or fixed specification of the allocation of uplink capacity between the reservation channel and data channel does not offer the optimal system performance over the entire operating range from low to high utilization.

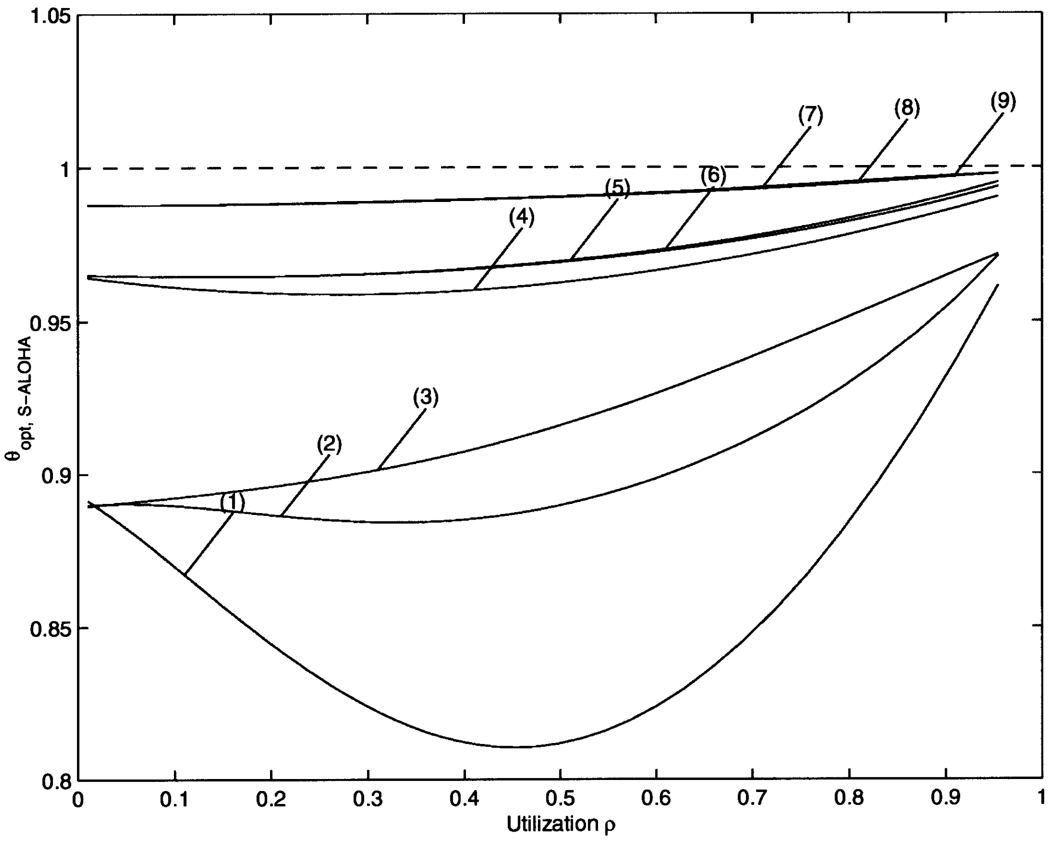


Figure B.5: Results for optimal capacity assignment $\theta_{\mathrm{opt,S-ALOHA}}$ for different α and β parameters: (1) $\alpha = 10^2$, $\beta = 10^{-4}$; (2) $\alpha = 10^2$, $\beta = 10^{-3}$; (3) $\alpha = 10^2$, $\beta = 10^{-2}$; (4) $\alpha = 10^3$, $\beta = 10^{-4}$; (5) $\alpha = 10^3$, $\beta = 10^{-3}$; (6) $\alpha = 10^3$, $\beta = 10^{-2}$; (7) $\alpha = 10^4$, $\beta = 10^{-4}$; (8) $\alpha = 10^4$, $\beta = 10^{-3}$; (9) $\alpha = 10^4$, $\beta = 10^{-2}$.

Bibliography

[1] Norman Abramson. Development of the ALOHANET. *IEEE Transactions on Information Theory*, 31:119–123, March 1985.

[2] Leonard Kleinrock and Simon S. Lam. Packet Switching in a Multiaccess Broadcast Channel: Performance Evaluation. *IEEE Transactions on Communications*, 4:410–422, April 1975.

[3] Tri T. Ha. *Digital Satellite Communications.* McGraw-Hill, 1990.

[4] John I. Capetanakis. Generalized TDMA: The Multi-Accessing Tree Protocol. *IEEE Transactions on Communications*, 27:1476–1484, October 1979.

[5] John I. Capetanakis. Tree Algorithms for Packet Broadcast Channels. *IEEE Transactions on Information Theory*, 25:505–515, September 1979.

[6] Robert G. Gallager. A Perspective on Multiaccess Channels. *IEEE Transactions on Information Theory*, 31:124–142, March 1985.

[7] Dimitri Bertsekas and Robert Gallager. *Data Networks.* Prentice Hall, 1992.

[8] Leonard Kleinrock and Fouad A. Tobagi. Packet Switching in Radio Channels: Part I–Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics. *IEEE Transactions on Communications*, 23:1400–1416, December 1975.

[9] Lawrence G. Roberts. Dynamic Allocation of Satellite Capacity Through Packet Reservation. In *1973 National Computing Conference, AFIPS Conference Proceedings*, volume 42, pages 711–716, April 1974.

[10] Fouad A. Tobagi and Leonard Kleinrock. Packet Switching in Radio Channels: Part III–Polling and (Dynamic) Split-Channel Reservation Multiple Access. *IEEE Transactions on Communications*, 24:832–845, August 1976.

[11] D. Raychaudhuri. Performance Analysis of Random Access Packet-Switched Code Division Multiple Access Systems. *IEEE Transactions on Communications*, 29:895–901, June 1981.

[12] Atul C. Khanna. An Analysis of Multiaccess Reservation Strategies for Satellite Channels. Master's thesis, Massachusettes Institute of Technology, 1985.

[13] Leonard Kleinrock. *Queueing Systems, Volume I: Theory.* John Wiley & Sons, 1975.

[14] Wolfgang Fischer and Kathleen Meier-Hellstern. The Markov-modulated Poisson Process (MMPP) Cookbook. *Performance Evaluation*, 18:149–171, January 1992.

[15] Ronald W. Wolff. Poisson Arrivals See Time Averages. *Operations Research*, 30:223–231, March 1982.

[16] David M. Lucantoni. New Results on the Single Server Queue with a Batch Markovian Arrival Process. *Communication Statistics–Stochastic Models*, 7:1–46, January 1991.

[17] David M. Lucantoni, Kathlen S. Meier-Hellstern, and Marcel F. Neuts. A Single-Server Queue with Server Vacations and a Class of Non-Renewal Arrival Processes. *Advanced Applied Probability*, 22:676–705, January 1990.

[18] Marcel F. Neuts. *Matrix-Geometric Solutions in Stochastic Models–An Algorithmic Approach.* John Hopkins University Press, 1981.

[19] David M. Lucantoni and V. Ramaswami. Efficient Algorithms for Solving the Non-Linear Matrix Equations Arising in Phase Type Queues. *Communication Statistics–Stochastic Models*, 1:29–51, January 1985.

[20] Harry Heffes. A Class of Data Traffic Processes–Covariance Function Characterization and Related Queuing Results. *Bell System Technical Journal*, 59:897–929, July 1980.

[21] Harry Heffes and David M. Lucantoni. A Markov Modulated Characterization of Packetized Voice and Data Traffic and Related Statistical Multiplexer Performance. *IEEE Journal on Selected Areas in Communications*, 6:856–868, September 1986.

[22] Kathleen S. Meier-Hellstern. The Analysis of a Queue Arising in Overflow Models. *IEEE Transactions on Communications*, 37:367–372, April 1989.

[23] H. Sitaraman. Approximation of Some Markov-Modulated Poisson Processes. *ORSA Journal on Computing*, 3:12–22, December 1991.

[24] Andrea Baiocchi and Nicola Blefari-Melazzi. Steady-State Analysis of the MMPP/G/1/K Queue. *IEEE Transactions on Communications*, 41:531–534, April 1993.

[25] V. Ramaswami. The N/G/1 Queue and its Detailed Analysis. *Advanced Applied Probability*, 12:222–261, January 1980.

[26] Marcel F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications.* Marcel Dekker, 1989.

[27] B. Meini. New Convergence Results on Functional Iteration Techniques for the Numerical Solution of M/G/1 Type Markov Chains. *Numerische Mathematik*, 78:39–58, January 1997.

[28] David M. Lucantoni. The BMAP/G/1 Queue: A Tutorial. In *Performance Evaluation of Computer and Communication Systems: Joint Tutorial Papers of Performance 1993 and Sigmetrics 1994*, volume 1, pages 330–358, May 1993.

[29] Victor S. Frost and Benjamin Melamed. Traffic Modeling For Telecommunications Networks. *IEEE Communications Magazine*, 3:70–81, March 1994.

[30] Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*, 2:1–15, February 1994.

[31] Riccardo Gusella. Characterizing the Variability of Arrival Processes with Indexes of Dispersion. *IEEE Journal on Selected Areas in Communications*, 9:203–210, February 1991.

[32] S. Ben Slimane and T. Le-Ngoc. A Doubly Stochastic Poisson Model for Self-Similar Traffic. In *1995 IEEE International Conference on Communications - Gateway to Globalization*, volume 1, pages 456–460, June 1995.

[33] J.H.B. Deane, C. Smythe, and D.J. Jefferies. Self-Similarity in a Deterministic Model of Data Transfer. *International Journal on Electronics*, 90:677–691, January 1996.

[34] Andrew R. Cohen, Jerrold A. Heller, and Andrew J. Viterbi. A New Coding Technique for Asynchronous Multiple Access Communication. *IEEE Transactions on Communication Technology*, 19:849–855, October 1971.

[35] Robert K. Morrow and James S. Lehnert. Packet Throughput in Slotted ALOHA DS/SSMA Radio Systems with Random Signature Sequences. *IEEE Transactions on Communications*, 40:1223–1230, July 1992.

[36] R.C. Sommer. On the Optimization of Random Access Discrete Address Communications. *Proceedings of the IEEE*, 52:1255, October 1964.

[37] D. Chesler. Performance of a Multiple Address RADA System. *IEEE Transactions on Communication Technology*, 14:369–372, August 1966.

[38] Andrew J. Viterbi. Orthogonal Tree Codes for Communication in the Presence of White Gaussian Noise. *IEEE Transactions on Communication Technology*, 15:238–242, April 1967.

[39] Andrew J. Viterbi. Very Low Rate Convolutional Codes for Maximum Theoretical Performance of Spread-Spectrum Multiple-Access Channels. *IEEE Journal on Selected Areas in Communications*, 8:641–649, May 1990.

[40] James S. Lehnert and Michael B. Pursley. Error Probabilities for Binary Direct-Sequence Spread-Spectrum Communications with Random Signature Sequences. *IEEE Transactions on Communications*, 35:87–98, January 1987.

[41] Brian D. Woerner and Wayne E. Stark. Improved Upper Bounds on the Packet Error Probabiliy of Slotted and Unslotted DS/SS Systems. *IEEE Transactions on Communications*, 43:3055–3062, December 1995.

[42] Dimitrios Makrakis and K.M. Sundara Murthy. Spread Slotted ALOHA Techniques for Mobile and Personal Satellite Communication Systems. *IEEE Journal on Selected Areas in Communications*, 10:985–1002, August 1992.

[43] Donald H. Davis and Steven A. Gronemeyer. Performance of Slotted ALOHA Random Access with Delay Capture and Randomized Time of Arrival. *IEEE Transactions on Communications*, 28:703–710, May 1980.

[44] Ramjee Prasad, Richard D.J. van Nee, and Rogier N. van Wolfswinkel. Performance Analysis of Multiple Access Techniques for Land-Mobile Satellite Communications. In *1994 IEEE GLOBECOM. Communications: The Global Bridge*, volume 2, pages 740–744, November 1994.

[45] Andreas Polydoros and John Silvester. Slotted Random Access Spread Spectrum Networks: An Analytical Framework. *IEEE Journal on Selected Areas in Communications*, 5:989–1002, July 1987.

[46] Sheldon M. Ross. *Introduction to Probabilty Models.* Boston : Academic Press, 1989.

[47] Leonard Kleinrock. *Communication Nets: Stochastic Message Flow and Delay.* McGraw-Hill Book Company, 1964.

[48] Fouad A. Tobagi and Leonard Kleinrock. Packet Switching in Radio Channels: Part III–Polling and (Dynamic) Split-Channel Reservation Multiple Access. *IEEE Transactions on Communications*, 24:832–844, August 1976.