

# A Multichannel Random Access Protocol for Multislot Short Messages with Delay Constraints

Xiaoming Zhou and John S. Baras  
Institute of System Research  
University of Maryland at College Park, MD, 20742  
Email: xmzhou@isr.umd.edu

**Abstract**—A multichannel random access protocol called FMCSA for multislot short messages is proposed in this paper to be used in access networks with large propagation delay. This protocol combines random access with the use of packet level forward error correction coding for new messages and scheduled retransmissions for partially received messages. Analytical and simulation results show that FMCSA can achieve a higher throughput and lower delay than slotted Aloha. When the system is operating in the low load region, the short messages can be delivered in their first attempts with very high probability. With the load increasing, more messages will be received partially in their first attempts and the scheduled retransmission scheme will guarantee the partially received messages to be recovered in their second attempts. Therefore the delay performance of FMCSA is much more robust to the load fluctuation than slot Aloha.

## I. INTRODUCTION

We consider short message transfer in the reverse channel over a star satellite network. Short messages such as HTTP requests have a typical size about 400 bytes [10][5][3][19]. At the MAC layer, a short message is fragmented into multiple smaller MAC packets. For example, assume each MAC layer slot carries a MAC header of 5 bytes and a MAC payload of 48 bytes, therefore a short message of 432 bytes will be segmented into 9 MAC packets. Only after all the MAC packets of a message are received will the hub reassemble the message and forward it to the upper layer.

All the terminals and the hub are synchronized and MAC packets are allowed to be sent only at the beginning of a time slot. We assume there are no other errors except the collisions caused by more than one MAC packets sent in the same slot. Whenever there is a collision, all the MAC packets involved are destroyed. A selective reject retransmission strategy is used for collision recovery in which only the collided MAC packet will be retransmitted. All new messages generated by the upper layer are buffered at the MAC layer queue

even when a terminal is backlogged and is attempting to retransmit a previous MAC packet [4].

We are interested in a wideband reverse channel which serves a large number of terminals. With the increase of the bandwidth, the transmission time per bit is decreased therefore the peak power of the transmitter has to be increased to keep the energy per bit constant [1]. However safety and cost constraints set an upper limit on the peak power of a terminal transmitter. A straightforward way to extend a narrowband channel to wideband can be done by dividing the wideband channel into a number of narrowband channels in frequency domain and operating in the MF-TDMA format. We assume each terminal has only one transmitter which could possibly hop to another narrowband channel after it finishes transmitting a MAC packet in its current channel. In the above we assume the terminals are peak power constrained which puts an upper limit on the transmission rate of each narrowband channel. We further assume that the terminals are not average power constrained and they can keep on sending as long as there are packets in their MAC layer queues.

The forward channel from the hub to the terminals uses a different frequency from those used in the reverse channel i.e. FDD and it is operated in a TDM fashion. Each terminal filters the received packets based on its own MAC address and only delivers those destined to it. The propagation delay between each terminal and the hub is 250ms.

Multichannel slotted Aloha (MCSA) has been proposed by Birk and Keren to be used in the reverse channel [4]. With multiple channels, immediate retransmission following a collision is permitted by randomly choosing a channel. While in single channel slotted Aloha, temporal random retransmission is the only option to avoid a definite repeated collision. For short transaction based messages, it is desirable to deliver them with the smallest time delay possible. However in multichannel slotted Aloha, the success probability of a  $k$

slot message in its first attempt is  $e^{-G*k}$ , where  $G$  is the traffic load. If we assume all the channels are operating at the same low load of  $G = 0.1$  and message length equals to nine slots, the success probability of first attempts is only 0.41. This means that only 41% of the messages can be delivered in the first try i.e. with propagation delay of 250ms. While 59% of the messages will incur retransmissions and the message delay will be at least 750ms. Therefore in a GEO satellite network which has large propagation delay, the multichannel slotted Aloha cannot deliver the multislot short messages in a timely manner.

In the above the narrowband slotted Aloha is extended to wideband in the frequency domain. It is worth to mention that it is also possible to extend narrowband Aloha to wideband in the code domain. Spread Aloha [1] uses a common code for all the terminals and separates different terminals by a random timing mechanism. Spread Aloha is equivalent to the FDMA/Aloha approach in the same way of creating separate logical channels [1]. Here we focus on the FDMA/Aloha approach and we argue that the performance of the FDMA/Aloha approach should be similar to Spread Aloha.

In addition to the random access protocols, reservation based protocols are also proposed in the literature. one protocol called combined free demand assignment multiple access (CFDAMA) [8] introduces the new concept of free bandwidth assignment. CFDAMA first allocates reverse channel bandwidth to the terminals on a demand basis. However when there is no demand, the scheduler allocates the remaining free bandwidth to the terminals according to some scheduling schemes such as round robin. The reverse channel bandwidth is managed by a centralized scheduler located at the hub on the ground. CFDAMA is efficient for bulk data transfers. However for bursty short transfers, due to the very large terminal population, the probability for a terminal to receive a free allocation is pretty low. Therefore most of the time a bandwidth request has to be sent which introduces additional delay of about 500ms.

From above, we can see the proposed protocols do not perform well in term of efficiency and delay when driven by bursty short messages. This motivates us to design a new MAC protocol which can deliver short messages with small delay and reasonable efficiency. Fixed assignment is ruled out of our consideration because it is not efficient for the bursty traffic. Reservation based protocols introduce the request phase overhead which is too expensive for short messages. Our protocol is a random access protocol which can improve the delay

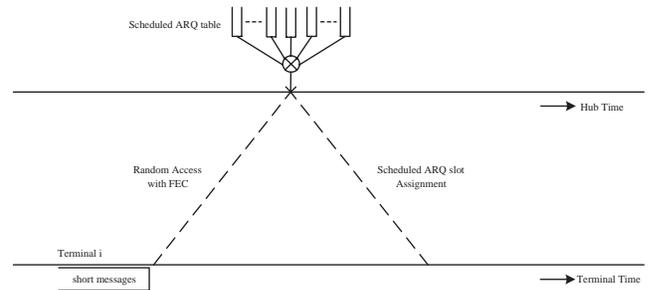


Fig. 1. System model of fast multichannel slotted Aloha

performance of multislot short messages dramatically compared with the aforementioned protocols.

The rest of this paper is organized as follows. Section II describes the FMCSA protocol. An analytical model for FMCSA is proposed in section III to calculate the system throughput and retransmission rate. Section IV gives the simulation results of delay performance in FMCSA for different system parameters such as FEC code rate, channel bandwidth, terminal population and message arrival patterns. Section V presents how FMCSA can be used to handle different slot sizes and heterogeneous message lengths. Processing overhead introduced in FEC coding is also discussed in section V. Finally, Section VI concludes this paper.

## II. FAST MULTICHANNEL SLOTTED ALOHA

Our protocol called fast multichannel slotted Aloha (FMCSA) is based on multichannel slotted Aloha (MCSA) with new features such as packet level FEC and scheduled ARQ designed specifically for networks with large propagation delay. The system model of our protocol is shown in figure 1. In this paper, we assume one slot can carry exactly one MAC packet.

When a new message arrives from the upper layer, it is first segmented into  $k$  MAC packets. Then the  $k$  packets are encoded into  $n$  code packets [2][11][18][15][14]. The first  $k$  code packets contain the original message. The remaining  $n - k$  are parity packets. The  $n$  code packets are sent in  $n$  consecutive time slots and each slot is randomly chosen from the remaining FDMA Aloha channels which have not been reserved for the retransmissions. When the hub schedules the retransmissions, it makes sure that for each slot there is at least one channel left for random access. For every  $n$  code packets, if any  $k$  or more out of them are received, the original message can be recovered correctly from the erasure. Each of the  $n$  code packets carries a unique message id number and a sequence number in its header. There are three possible

outcomes of a message after its first attempt.

In the first case, the message is fully received. In this case, the number of packets received correctly  $m$  is no less than  $k$  so that the message can be reassembled and forwarded to the upper layer. A positive acknowledgement for this message will be sent to the terminal as soon as  $m$  becomes equal to  $k$ . It is possible that additional parity packets of the message will arrive after this. Because the whole message has already been received correctly, the additional parity packets will simply be discarded by the hub.

In the second case where the message is partially received i.e.  $m$  is less than  $k$  but greater than zero, packet retransmission becomes necessary. A selective reject strategy [17] [16] is employed in FMCSA for packet recovery. From the sequence numbers and the message id numbers carried in the MAC headers, the hub can figure out which packets are collided in the message and it will reserve  $k-m$  slots for the message recovery rather than let the terminal to retransmit the collided packets in the random access mode [20]. Because of the scheduled ARQ, FMCSA can guarantee the successful delivery of a message in its second attempt as long as there is at least one packet getting through in its first attempt. For example, a three-slot message is encoded into five code packets and they are sent in five consecutive slots. If only the second and the fourth code packets are received correctly, the hub will find out that the message length is three and it needs one more code packets to reassemble the message. It then allocates one slot for this message. After the terminal receives the allocation, it sends an additional code packet such as the first code packet in the reserved slot so that the hub will be able to recover the message after it receives this packet.

In the worst case where none of the  $n$  code packets get through i.e. the whole message is erased, the terminal will timeout and the collided  $n$  code packets will be sent again in the same way a new message is transmitted.

### III. PERFORMANCE ANALYSIS

In this section, we model FMCSA and analyze its throughput and delay performance. We will show that FMCSA can achieve a higher maximum throughput than MCSA. For the same system throughput, FMCSA can deliver the short messages in the first attempts with much higher probability than MCSA. With the increasing of system load, the message delay performance is still acceptable and it degrades much more gracefully than MCSA.

#### A. System Throughput

The system throughput we are interested in is the throughput seen in the upper layer i.e. the effective throughput of the short messages which does not include the throughput of the parity packets. In the analysis, we assume that the new MAC packet arrival rate in each of the  $N$  parallel channels is Poisson distributed with mean  $\lambda_k$ . Therefore after FEC coding, the arrival rate is increased to  $\lambda_n = \frac{n}{k} * \lambda_k$ . We classify the retransmissions into two cases. In the first case, a  $n$  slot message is partially received and scheduled ARQ is used for its retransmission. We assume that the arrival rate of scheduled ARQ assignments in each channel is Poisson distributed with mean  $\lambda_r$ . In the second case, none of the  $n$  MAC packets in a message is received and all of them have to be sent again. We assume its arrival rate is also Poisson distributed with mean  $\lambda_{nr}$ .

The system throughput  $S$  contains two parts. One is contributed by the  $n$  slot messages which include new message transmissions and totally erased message retransmissions  $S(\lambda_n, \lambda_{nr}, \lambda_r)$ . Note that it is also a function of the scheduled retransmission rate  $\lambda_r$ . This is because the  $n$  slot messages are sent in the channels left by the scheduled retransmissions. The other part of the system throughput is contributed by the less than  $k$  slot retransmissions i.e. the scheduled ARQ  $S(\lambda_r)$ . Because reservation is used in scheduled ARQ, it is guaranteed that all such retransmissions will be received correctly i.e.

$$S(\lambda_r) = \lambda_r \quad (1)$$

Because the channels reserved for scheduled ARQ cannot be used by the  $n$  slot messages, it is equivalent that their arrival rates  $\lambda_n$  and  $\lambda_{nr}$  are both increased by a factor of  $1/(1-\lambda_r)$ . Let the total arrival rate of  $n$  slot messages be  $G_n$ , then we have

$$G_n = \frac{\lambda_n + \lambda_{nr}}{1 - \lambda_r} \quad (2)$$

where

$$\lambda_n = \frac{n}{k} * \lambda_k \quad (3)$$

It should be noted that not all the correctly received code packets contribute to the effective throughput  $S$ . Let the number of correctly received packets in a single  $n$  slot message be  $m$ . If  $0 \leq m \leq k$ , all the  $m$  received packets contribute to the effective throughput  $S$ . On the other hand, if  $m > k$ , only  $k$  out of  $m$  code packets

contribute to  $S$ . The probability of  $m$  code packets received in a  $n$  slot message is

$$P_r(m) = \binom{n}{m} \cdot (1 - e^{-G_n})^{n-m} \cdot e^{-G_n \cdot m} \quad (4)$$

The probability of an arbitrary received code packet in  $n$  slot messages which contributes to  $S$  is

$$P_e = \frac{\sum_{m=0}^n \min(m, k) \cdot P_r(m)}{\sum_{m=0}^n m \cdot P_r(m)} \quad (5)$$

Therefore the throughput of the  $n$  slot messages is

$$S(\lambda_n, \lambda_{nr}, \lambda_r) = P_e \cdot (\lambda_n + \lambda_{nr}) \cdot e^{-G_n} \quad (6)$$

The effective system throughput is

$$S = S(\lambda_n, \lambda_{nr}, \lambda_r) + S(\lambda_r) \quad (7)$$

In steady state, the arrival rate of the new packets equals to the effective system throughput  $S = \lambda_k$ , then from equations 7, 6 and 1, we can get

$$\lambda_k = P_e \cdot (\lambda_n + \lambda_{nr}) \cdot e^{-G_n} + \lambda_r \quad (8)$$

Also note that the MAC packet arrival rate due to the totally erased message retransmissions is as following

$$\lambda_{nr} = (\lambda_n + \lambda_{nr}) \cdot (1 - e^{-G_n})^n \quad (9)$$

From equations 2 and 9, we can have

$$\lambda_n = (1 - \lambda_r) \cdot G_n \cdot (1 - (1 - e^{-G_n})^n) \quad (10)$$

$$\lambda_{nr} = (1 - \lambda_r) \cdot G_n \cdot (1 - e^{-G_n})^n \quad (11)$$

From equations 3 and 10, we can get

$$\lambda_k = \frac{k}{n} \cdot (1 - \lambda_r) \cdot G_n \cdot (1 - (1 - e^{-G_n})^n) \quad (12)$$

Substitute  $\lambda_k$  with the right side in equation 12 and  $(\lambda_n + \lambda_{nr})$  with  $(1 - \lambda_r) \cdot G_n$  into equation 8, we can get the following

$$\begin{aligned} \frac{k}{n} \cdot (1 - \lambda_r) \cdot G_n \cdot (1 - (1 - e^{-G_n})^n) \\ = P_e \cdot (1 - \lambda_r) \cdot G_n \cdot e^{-G_n} + \lambda_r \end{aligned} \quad (13)$$

Equation 13 relates  $G_n$  and  $\lambda_r$ . Therefore  $G_n$  can be calculated numerically for a given  $\lambda_r$ . Once  $G_n$  and  $\lambda_r$  are known, we can get the system throughput  $\lambda_k$  by using equation 12 as well as the  $n$  slot message retransmission rate  $\lambda_{nr}$  through equation 9.

Figure 2 shows the throughput performance of FMCSA and MCSA with respect to the total offered load  $G$ . In FMCSA,  $G = \lambda_n + \lambda_{nr} + \lambda_r$  where  $\lambda_n$  is the new code packet arrival rate. We can see from the figure that FMCSA can achieve a higher maximum throughput than

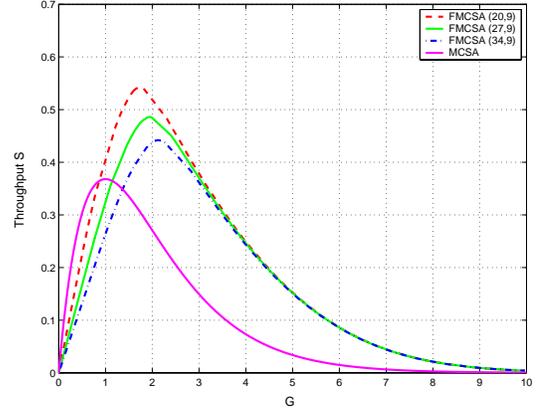


Fig. 2. The system throughput of FMCSA and MCSA

MCSA. The maximum throughput of MCSA is 0.368. While FMCSA with a FEC code (34,9) has a maximum throughput of 0.442. With the code rate increased to 9/27, the maximum throughput is increased to 0.486. When FEC code (20,9) is used, the maximum throughput becomes 0.542. FMCSA have the same bistable property as MCSA. We would like to operate the two protocols in the stable region, i.e. on the left side of the maximum throughput point. Due to the parity packets sent in FMCSA, more load is offered to the channel than in MCSA to achieve the same throughput. However the figure does not give too much information about the message delay. In the next section will compare the delay performance of FMCSA and MCSA for a given throughput.

### B. First Attempt Success Probability

Because the propagation delay in satellite networks is very large, it is desirable that the messages can be received correctly in the first attempts. Otherwise one round trip time delay about 500ms will be introduced.

In the following, we calculate the first attempt success probability of FMCSA with a FEC code  $(n, k)$ . MCSA is equivalent to the degenerated case of FMCSA with  $n = k$  and no scheduled retransmissions. Let  $p$  be the probability with which a new MAC packet can get through the channel. Therefore,

$$p = e^{-G} \text{ in MCSA} \quad (14)$$

$$p = e^{-G_n} \text{ in FMCSA} \quad (15)$$

Where  $G$  is the total offered load in the MCSA channel and  $G_n$  is given by equation 2 which is the load offered to the channel left by the scheduled ARQ in FMCSA.

The full message received probability, i.e. the probability of no less than  $k$  packets received in the first attempt is

$$P_r(m \geq k) = \sum_{m=k}^n \binom{n}{m} \cdot (1-p)^{n-m} \cdot p^m \quad (16)$$

The probability of none of the  $n$  packets getting through is

$$P_r(m = 0) = (1-p)^n \quad (17)$$

The probability of partially received messages in the first attempts, i.e.  $0 < m < k$  is

$$P_r(0 < m < k) = 1 - P_r(m \geq k) - P_r(m = 0) \quad (18)$$

Numerical results are shown in table I and table II about the probability of the above three cases for different system throughput in MCSA and FMCSA. Here  $k = 9$  and a (27,9) error correction code is used ( $n = 27$ ) in FMCSA. From the tables, we can see that the successful probability in the first attempt decreases very fast in MCSA with the increase of the throughput. While in FMCSA, this probability is not sensitive to the throughput. Actually when the system throughput  $S$  is below 0.25, almost all the messages can get through in their first attempts. Even when the throughput is increased up to 0.3, the successful probability of a message in its first attempt is as high as 82.31% in FMCSA compared with only 1.22% in MCSA.

From table I and table II, we notice that the probability of none of the  $n$  code packets<sup>1</sup> getting through is pretty low for both protocols even when the throughput is high. This means almost all the messages are either fully received or partially received in the first attempt. Because the packets of a partially received message can reserve enough slots to retransmit the additional code packets in FMCSA, the message can be recovered in the second attempt. While MCSA does not take advantage of this fact, the retransmissions may again incur collisions. Therefore FMCSA can improve the delay performance significantly when compared with MCSA. Table III and table IV show similar results for FEC code (20,9) and (34,9) respectively. From above, we can see that the first attempt success probability in FMCSA is not sensitive to system throughput as well as FEC code rate.

<sup>1</sup> $n = 27, k = 9$  in FMCSA and  $n = 9, k = 9$  in MCSA

TABLE I  
NONE, PARTIAL AND FULL MESSAGE RECEIVED PROBABILITY IN  
MCSA WITH DEGENERATED FEC CODE (9,9)

$S$	$P_r(m = 0)$	$P_r(0 < m < k)$	$P_r(m \geq k)$
0.01	$1.045 * 10^{-18}$	0.0869	0.9131
0.05	$2.476 * 10^{-12}$	0.3777	0.6223
0.1	$1.658 * 10^{-9}$	0.6344	0.3656
0.15	$8.730 * 10^{-8}$	0.8012	0.1988
0.2	$1.687 * 10^{-6}$	0.9030	0.0970
0.25	$1.998 * 10^{-5}$	0.9599	0.0401
0.3	$1.947 * 10^{-4}$	0.9876	0.0122
0.35	$2.400 * 10^{-3}$	0.9960	0.0016
0.4	N/A	N/A	N/A

TABLE II  
NONE, PARTIAL AND FULL MESSAGE RECEIVED PROBABILITY IN  
FMCSA WITH FEC CODE (27,9)

$S$	$P_r(m = 0)$	$P_r(0 < m < k)$	$P_r(m \geq k)$
0.01	$\simeq 0$	$\simeq 0$	$\simeq 1$
0.05	$\simeq 0$	$\simeq 0$	$\simeq 1$
0.1	$1.470 * 10^{-16}$	$1.684 * 10^{-6}$	0.9999
0.15	$1.251 * 10^{-12}$	$3.261 * 10^{-4}$	0.9997
0.2	$4.652 * 10^{-10}$	$7.144 * 10^{-3}$	0.9929
0.25	$3.184 * 10^{-8}$	$4.896 * 10^{-2}$	0.9510
0.3	$9.134 * 10^{-7}$	$1.769 * 10^{-1}$	0.8231
0.35	$1.566 * 10^{-5}$	$4.156 * 10^{-1}$	0.5844
0.4	$3.029 * 10^{-4}$	$7.509 * 10^{-1}$	0.2488

TABLE III  
NONE, PARTIAL AND FULL MESSAGE RECEIVED PROBABILITY IN  
FMCSA WITH FEC CODE (20,9)

$S$	$P_r(m = 0)$	$P_r(0 < m < k)$	$P_r(m \geq k)$
0.01	$\simeq 0$	$\simeq 0$	$\simeq 1$
0.05	$\simeq 0$	$\simeq 0$	$\simeq 1$
0.1	$9.800 * 10^{-15}$	$9.827 * 10^{-5}$	0.9999
0.15	$1.126 * 10^{-11}$	$3.068 * 10^{-3}$	0.9969
0.2	$1.268 * 10^{-9}$	$2.417 * 10^{-2}$	0.9758
0.25	$4.174 * 10^{-8}$	$9.203 * 10^{-2}$	0.9080
0.3	$6.954 * 10^{-7}$	$2.297 * 10^{-1}$	0.7703
0.35	$8.517 * 10^{-6}$	$4.423 * 10^{-1}$	0.5577
0.4	$1.106 * 10^{-4}$	$7.129 * 10^{-1}$	0.2870

TABLE IV

NONE, PARTIAL AND FULL MESSAGE RECEIVED PROBABILITY IN FMCSA WITH FEC CODE (34,9)

$S$	$P_r(m=0)$	$P_r(0 < m < k)$	$P_r(m \geq k)$
0.01	$\simeq 0$	$\simeq 0$	$\simeq 1$
0.05	$\simeq 0$	$\simeq 0$	$\simeq 1$
0.1	$8.546 * 10^{-18}$	$9.026 * 10^{-8}$	0.9999
0.15	$4.257 * 10^{-13}$	$8.625 * 10^{-5}$	0.9999
0.2	$4.308 * 10^{-10}$	$4.311 * 10^{-3}$	0.9957
0.25	$5.575 * 10^{-8}$	$4.593 * 10^{-2}$	0.9541
0.3	$2.612 * 10^{-6}$	$2.103 * 10^{-1}$	0.7897
0.35	$6.095 * 10^{-5}$	$5.224 * 10^{-1}$	0.4775
0.4	$2.148 * 10^{-3}$	$9.014 * 10^{-1}$	0.0965

### C. Retransmission Rate and Operation Region

In MCSA, the average number of attempts for the successful reception of a single MAC packet is  $e^G$ . Therefore the average delay for MAC packet  $i$  denoted by  $D_{MCSA}(i)$  [7] is given as follows

$$D_{MCSA}(i) = 1 + t_{prop} + (e^G - 1)(1 + 2 * t_{prop}) \quad (19)$$

Where  $t_{prop}$  is the normalized propagation delay<sup>2</sup>. Because the  $t_{prop}$  is large in satellite networks, from equation 19 we can see that retransmissions will increase the packet delay dramatically as the load  $G$  increases. Therefore to deliver the packets with short delay, the MCSA channel bandwidth should be sized such that it operates in the low load region such as  $G = 0.1$  most of the time, in which every attempt including first time transmissions and retransmissions of a packet has a probability of 90.5% to get through the channel.

While in the above we calculate the delay for a single MAC packet, the whole  $k$  slot message delay is determined by the delay of the last received packet generated from the message. Therefore the message delay in MCSA is

$$D_{MCSA} = \max(D_{MCSA}(1), D_{MCSA}(2), \dots, D_{MCSA}(k-1), D_{MCSA}(k)) \quad (20)$$

Even operating in the low load region such as  $G = 0.1$ , the probability of the message getting through in the first attempt is only 41% as calculated in section I. This means to improve the message delay performance in MCSA, the channel may have to be operated in an even

<sup>2</sup>The propagation delay  $t_{prop}$  is normalized to the packet transmission time

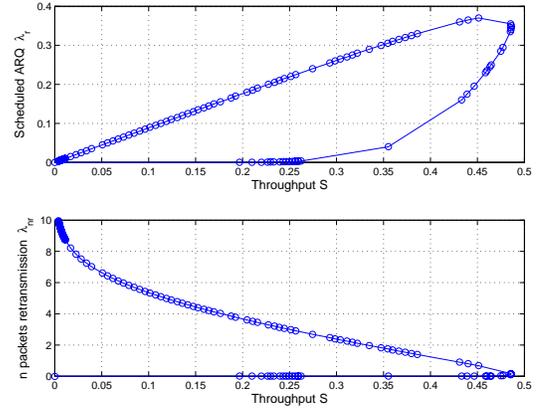


Fig. 3. The retransmission rate of FMCSA for different system throughput with FEC code (27,9)

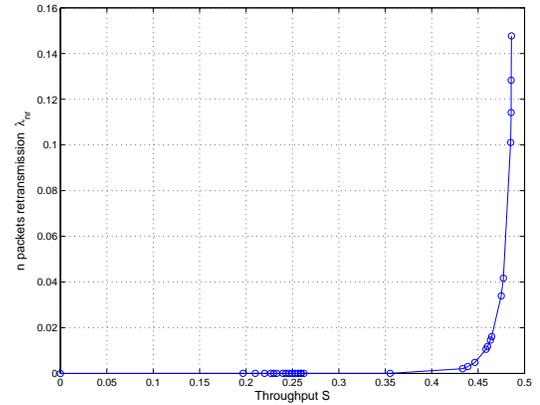


Fig. 4. The totally erased message retransmission rate of FMCSA for different system throughput with FEC code (27,9)

lower load region than  $G = 0.1$ , which of course is very inefficient in term of bandwidth utilization.

In the following we will show that FMCSA can operate much more efficiently without sacrificing the delay performance. The retransmissions in FMCSA include two parts: the scheduled ARQ and the totally erased message retransmissions. We plot these two retransmission rates with respect to the system throughput in figure 3, figure 5 and figure 6 for different code rates. Our discussion will focus on the (27, 9) code. The other two codes perform similarly to this code. Figure 3 shows the two retransmission rates at different operating points. The curve can be divided into two regions i.e. the stable region<sup>3</sup> and the unstable region. In the unstable region, we can see that both retransmission rates are relatively

<sup>3</sup>In the upper plot of figure 3, the stable region is the part below the line connecting  $(0, 0)$  and  $(S_{max}, \lambda_{sr}^*)$ . In the lower plot of figure 3, the stable region is the part below the line connecting  $(0, 0)$  and  $(S_{max}, \lambda_{rr}^*)$ .

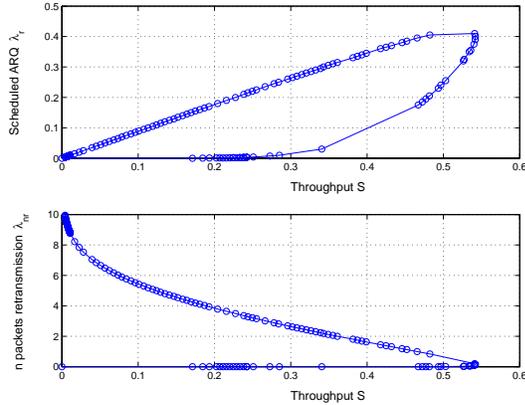


Fig. 5. The retransmission rate of FMCSA for different system throughput with FEC code (20,9)

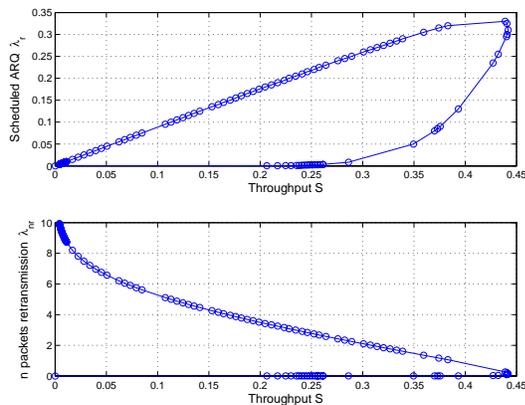


Fig. 6. The retransmission rate of FMCSA for different system throughput with FEC code (34,9)

large.

We would like to have FMCSA operate in the stable region. Figure 4 shows a more detailed  $n$  slot retransmission rate in the stable region. When the system throughput is less than 0.2, the scheduled ARQ rate  $\lambda_r$  and the  $n$  slot retransmission rate are both pretty close to zero. This means the messages can get through in their first attempts which confirms the results we get in table II. Therefore the delay performance of FMCSA is almost the same in region below  $S = 0.2$  and is much more robust to system load than that of MCSA.

When the system throughput  $S$  is between 0.2 and 0.4, we can see from the figure 4 that  $\lambda_{nr}$  is still very small and the scheduled ARQ rate begins to increase as shown in the upper plot of figure 3. This plot shows that as the system throughput increases, more messages will be delivered in the second attempts. From the above we can see that FMCSA should operate at throughput about 0.2 to get small message delay. While even under

temporary congestion, FMCSA can still be functional with acceptable message delay.

#### IV. SIMULATION RESULTS

In this section, we evaluate the delay performance of FMCSA and MCSA in OPNET. We will show that the simulation results with Poisson arrivals match very well with the analytical results we get in the previous section. We further evaluate the performance of FMCSA with Pareto arrivals. Our results show that the delay performance of FMCSA is not sensitive to the specific arrival distributions when the reverse channel is operating in the relatively low load region.

##### A. Delay Performance with Poisson Arrivals

In this section, we evaluate the delay performance of FMCSA with Poisson arrivals. There are 512 terminals in the network and all of them are sending short messages to the hub. The MAC packet size is 53 bytes with 5 bytes header. All the messages generated by the upper layer are of 9 slots i.e. 432 bytes. There are 25 parallel reverse channels and each has bandwidth of 64 kbps. Figure 7 shows the average message delay of FMCSA and MCSA for different throughput. The minimum message delay is achieved when the first 9 MAC packets of a message are received correctly at the hub and the delay is  $(250 + 9 \cdot 53 \cdot 8 / 64)$  i.e. 309.625 ms which is shown as the dash line in figure 7. We can see from the figure that when the system throughput is very small at around 1%, the average delay of both schemes are close to the minimum delay. While with the increase of the throughput, the average delay of MCSA increase dramatically. When the throughput is increase up to 35%, MCSA actually becomes saturated and the delay goes to infinity. As predicted by our performance analysis of FMCSA, its delay performance degrades much more gracefully than MCSA. Even when the system throughput is increased up to 20%, the average delay performance is comparable to the minimum delay. Figure 7 also shows that FMCSA can achieve a higher throughput than MCSA which makes it more robust to load fluctuation. For different code rate, there is not too much difference for the average delay when operating at the relatively low load region such as throughput less than 25%. In practice, this property gives us the flexibility to choose different codes.

Figure 8 shows the instantaneous message delay performance of FMCSA with FEC code (27,9) for three different throughput. If a new message arrives at an empty MAC layer queue and it is received correctly in the first attempt, the minimum delay is 309.625 ms

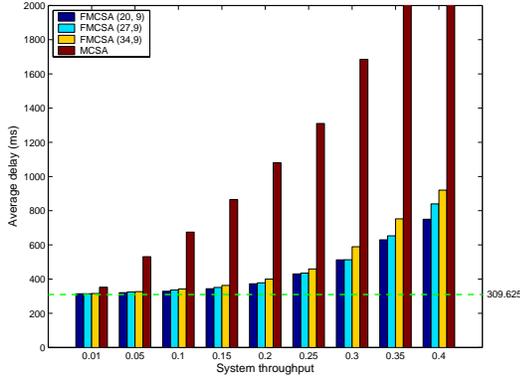


Fig. 7. The average message delay of FMCSA and MCSA for different throughput. There are 512 terminals and 25 parallel reverse channels. Each channel has bandwidth of 64kbps.

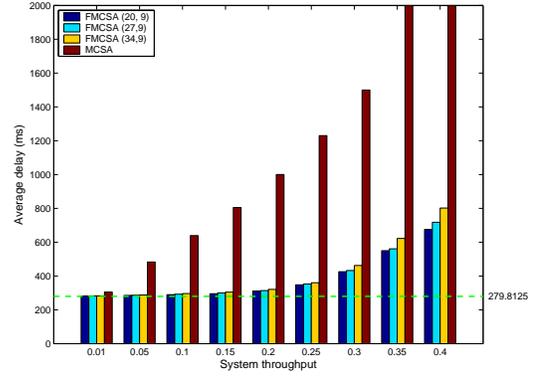


Fig. 9. The average message delay of FMCSA and MCSA for different throughput. There are 1024 terminals and 50 parallel reverse channels. Each channel has bandwidth of 128kbps.

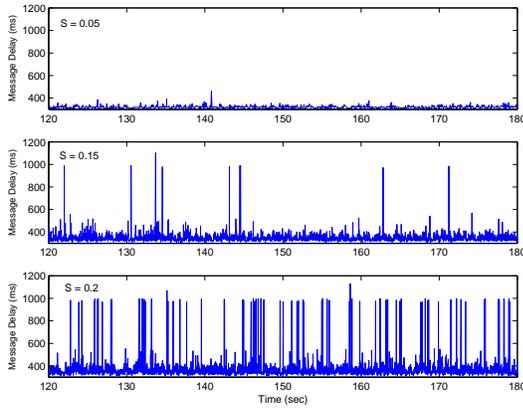


Fig. 8. The message delay of FMCSA (27,9) for different throughput. There are 512 terminals and 25 parallel reverse channels. Each channel has bandwidth of 64kbps.

as calculated in the above paragraph and similarly we can get the maximum delay is  $(250 + 27 \cdot 53 \cdot 8 / 64)$  i.e. 428.875 ms. Therefore if a message is not received fully in its first attempt, the message delay should be more than  $(428.875 + 2 \cdot 250)$  i.e. 928.875ms. So if the delay of a message is less than 928.875 ms, it must have been received in the first attempt. In topmost plot of figure 8, we can see that the messages are delivered in the first attempts when the throughput is 5%. When the throughput is increased to 15%, very few of them will incur retransmissions. We should point out here that because we allow multiple messages to be buffered at the terminals. If a new message arrives at the MAC layer queue and the terminal has not finished transmitting the previous packets. Under such circumstance even if the new message can be received in its first attempt, its message delay could be more than 428.875 ms, which corresponds to the case when a message arrives at an

empty MAC queue thus without any additional queuing delay<sup>4</sup>. This is confirmed by the middle plot in figure 8. When the throughput is increased further, we can see more messages will have to be retransmitted in the second attempts. We also calculate the first attempt success probabilities and compare them with our analysis in section III-B. As shown in table V, the simulation results match the analytical results very well.

We also evaluate the delay performance of FMCSA and MCSA with more terminals and higher bandwidth. Figure 9 shows the average delay of FMCSA and MCSA with 1024 terminals and 50 parallel reverse channels. Each channel has a bandwidth of 128kbps, therefore the total reverse channel bandwidth is 6.4Mbps. Because of the increase of the bandwidth, the transmission time of a 9 slot message is decreased to  $(9 \cdot 53 \cdot 8 / 128)$  i.e. 29.8125ms. In both FMCSA and MCSA, the minimum message delay is achieved when all of the first 9 slot MAC packets are received correctly in the first attempt and it is the transmission time plus the propagation delay (i.e.  $29.8125 + 250 = 279.8125$ ms) as shown in figure 9. Compared with the minimum delay of 309.625ms when each channel bandwidth is 64kbps, the increase of the bandwidth reduces the minimum delay by 29.8125ms. We can see in figure 9 that the average delay of FMCSA is very close to the minimum delay when the throughput is relatively small. With the increase of throughput, the delay performance of FMCSA degrades much more gracefully than that of MCSA. Actually figure 9 and figure 7 shows very similar results which gives us the evidence that FMCSA can scale to larger networks and higher channel bandwidth.

<sup>4</sup>Because a packet can only be sent at the beginning of a slot, the packet may still incur some queuing delay of less than one slot

TABLE V

THE ANALYTICAL AND SIMULATION RESULTS OF THE FIRST ATTEMPT SUCCESS PROBABILITY IN FMCSA WITH A FEC CODE (27,9)

$S$	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4
<i>Analysis</i>	$\simeq 1$	0.9999	0.9997	0.9929	0.9510	0.8231	0.5844	0.2488
<i>Simulation</i>	$\simeq 1$	$\simeq 1$	0.9979	0.9767	0.9068	0.7875	0.5674	0.2735

From above, we can see that FMCSA provides a system designer two dimensions of freedom to add more terminals to the network while keeping the delay performance the same. The network dimensioning can be done simply by adding more parallel channels while remaining the channel bandwidth as before. Another option is to leave the total number of channels unchanged however increase the bandwidth of each channel. As mentioned before, there is a limitation of each channel's peak power so that its bandwidth should not exceed some threshold. Therefore under some circumstance it requires the third design option which increases the number of channels and the bandwidth of each channel at the same time as we do in this experiment.

### B. Delay Performance with Pareto Arrivals

In the previous section, the message arrival rate follows the Poisson distribution. In this section, we evaluate the performance of FMCSA when the message arrival pattern is Pareto distributed [6][13]. Pareto distribution is the simplest heavy-tailed distribution with probability density function

$$p(x) = \alpha k^\alpha x^{-\alpha-1} \quad \text{where } \alpha, k > 0, x \geq k$$

and cumulative distribution function

$$F(x) = P[X \leq x] = 1 - (k/x)^\alpha$$

The parameter  $k$  is the location parameter and it represents the possible smallest value of random variable  $X$ . If  $\alpha \leq 2$ , it has infinite variance; if  $\alpha \leq 1$ , it has infinite mean. For  $1 < \alpha < 2$ , the mean of Pareto distribution is  $\alpha/(\alpha-1) * k$ . The Pareto distribution is hyperbolic over the entire range and as  $\alpha$  decreases, an arbitrary large portion of the probability mass could be present in the tail of the distribution [6]. Therefore a Pareto distributed random variable can generate extremely large values with nonnegligible probability.

Figure 10 shows the cumulative distribution of the message delay for Poisson and Pareto arrivals. There are 512 terminals and 25 parallel reverse channels in the network. Each channel has a bandwidth of 64kbps and FEC code (27,9) is used. In the Pareto distribution,  $\alpha$

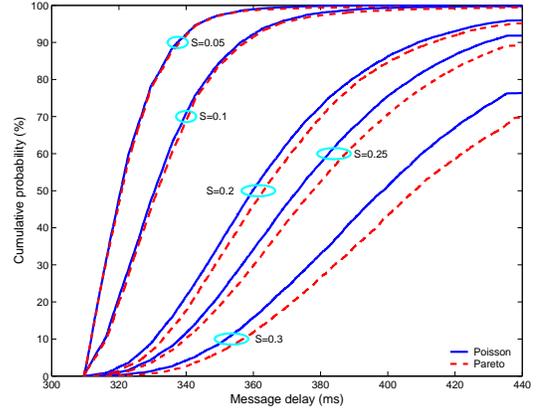


Fig. 10. The cumulative distribution of message delay in FMCSA (27, 9) with Poisson and Pareto arrivals. There are 512 terminals and 25 parallel reverse channels in the network. Each channel has bandwidth of 64kbps.

equals 1.5. Figure 10 shows that when the throughput is up to 20%, the delay distributions for both Poisson and Pareto arrivals are pretty close to each other. While the throughput is increased above 25%, the mean delay of Pareto arrivals is higher than Poisson arrivals due to the greater burstiness of the traffic. From this figure, we can see that the delay performance of FMCSA is robust to the arrival distributions when it is operated at the relatively low load region, which allows us to use the more mathematically tractable Poisson traffic model to predict the system performance.

## V. DEPLOYMENT CONSIDERATIONS

In this section, we will consider how FMCSA performs for the different system parameters such as different slot sizes and how FMCSA can be extended to handle heterogeneous message lengths. Finally, we will discuss the processing overhead introduced by FMCSA.

### A. Handling Homogeneous Message Lengths with Different Slot Sizes

In the previous sections, we evaluate the performance of FMCSA for nine-slot messages. However when either the message length changes or a different slot size has

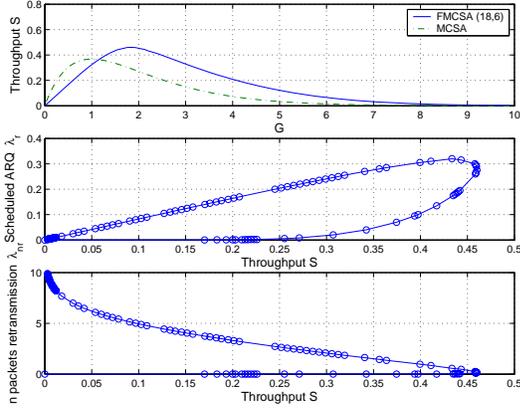


Fig. 11. The throughput and retransmission rates of FMCSA with FEC code (18,6).

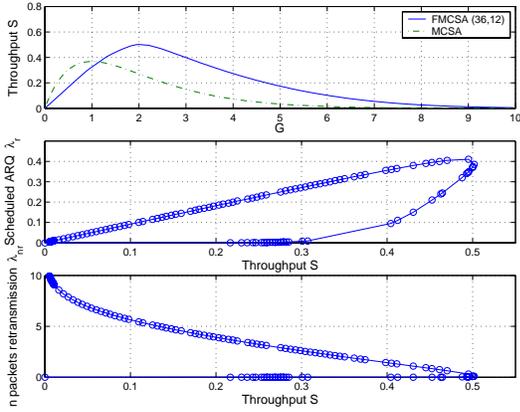


Fig. 12. The throughput and retransmission rates of FMCSA with FEC code (36,12).

been chosen, the number of MAC packets after the fragmentation of a message will change correspondingly. For example, in the previous sections, the message length is assumed to be 432 bytes and each slot can carry 48 byte payload. However if a larger slot size is chosen such that each slot can carry 72 byte payload, a 432-byte message will be fragmented into six MAC packets. On the other hand, if a smaller slot which can carry only 36 byte payload is used, a typical 432-byte message will generate twelve MAC packets after fragmentation. Figure 11 and figure 12 show the throughput and retransmission rates for six-slot and twelve-slot messages. From the topmost plots of both figures, we can see that FMCSA can achieve higher maximum throughput than MCSA for both cases. For the six-slot messages, FEC code (18,6) is used and for the twelve-slot messages, FEC code (36,12) is used. Actually the maximum throughput is 0.46 for FMCSA (18,6) and is 0.5 for FMCSA (36,12). The throughput results we show here is similar to the results

in figure 2 which shows the throughput performance of FMCSA for nine-slot messages.

From the middle and the lowest plots in figure 11 and figure 12, we can see that the retransmission rates are close to zero when the system throughput is less than 20% which means that almost all the messages can get through in their first attempts. This property is also very similar to the nine-slot message case. Therefore these results show that the performance of FMCSA is robust to the system parameters such as message length and slot size.

### B. Handling Heterogeneous Message Lengths

In the previous sections, we evaluate the performance of FMCSA when all the messages have the same length. In practice, the message lengths could be heterogeneous and follow certain distributions [10]. For example, if the message lengths are distributed uniformly in the region from 384 bytes to 480 bytes and each slot can carry 48 byte payload, the number of MAC packets after fragmentation will be either eight, nine or ten. Although the message lengths are different, we can still use the same FEC code such as (30,10). In case the message is less than 10 slots, the FEC code can be shortened by conceptually making a number of zero data symbols at the encoder. The zero symbols will not be transmitted, however they will be re-inserted at the hub for the decoding. For example, for a 8-slot message, the encoder can conceptually add two zero slots and generates a (30,10) codeword. However it transmits only the original eight data packets and certain number of parity packets.

Figure 13 shows the delay performance of FMCSA and MCSA for heterogeneous message lengths. In the upper plot, the message lengths are randomly chosen from eight, nine and ten slots with the same probability. As mentioned in the previous paragraph, the same FEC code (30,10) is used for all the messages and the number of additional parity packets is 16, 18 and 20 for 8, 9, 10 slot messages respectively. We can see from this plot that FMCSA outperforms MCSA even when the message lengths are heterogeneous. The upper plot also shows the delay performance of FMCSA when all the messages have the same length of nine slots. We can see from this plot that the average delay of FMCSA with heterogeneous message lengths is close to the homogeneous case. The lower plot shows the results for a larger message length range which is uniformly distributed from 7 to 11 slots. If each slot can carry 48 byte payload, the message lengths are in the range from 336 bytes to 528 bytes. It shows similar delay

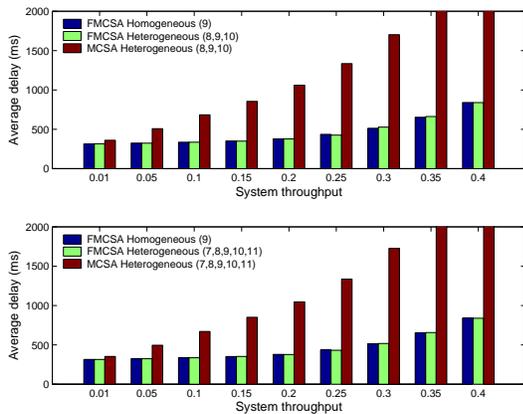


Fig. 13. The average message delay of FMCSA and MCSA for heterogeneous message length. There are 512 terminals and 25 parallel reverse channels. Each channel has bandwidth of 64kbps.

performance in FMCSA as the upper plot in which the message lengths are distributed in a smaller range.

### C. Processing Overhead

In FMCSA, we introduce packet level FEC. Compared to MCSA, FMCSA requires more processing to do the FEC coding/decoding. A Reed-Solomon erasure (RSE) correcting code as described in [11][18] can be used to generate the parity packets. Each terminal employs an  $(n, k)$  RSE code over a Galois field  $GF(2^m)$ . The symbol size  $m$  is picked to be sufficiently large such that  $n < 2^m$  [15]. However, it is difficult to implement a RSE coder to operate on symbols of the message size which is typically be on the order of several thousand bits. Let the message length be  $l \cdot m$  bits, where  $l$  is an integer. A multiple parallel RSE coding can be performed for each  $m$ -bit symbol in each data packet [15]. For example, RSE coding is performed on the first  $m$ -bit symbol in each of the  $k$  data packets such that  $n - k$   $m$ -bit parity symbols can be obtained. This process is then repeated for the rest  $l - 1$  symbols in each data packets to obtain the  $n - k$  parity packets.

In the  $n$  code packets, the first  $k$  packets are generated from the segmentation of the original message. If all of the  $k$  data packets are received, no decoding is required at the receiver. On the other hand, if  $j < n - k$  out of the  $k$  data packets are lost, the decoding overhead is proportional to  $j$  [15]. Nonnenmacher [15] evaluated the throughput of a software RSE codec developed by Rizzo [18] on a Pentium PC 133. It was shown that the coding and decoding throughput are on the order of 10 Mbps for  $k = 7$ , which is enough for our purposes. Therefore with more powerful machines and more efficient codec

algorithms, the processing of RSE FEC in FMCSA should not become the system bottleneck.

## VI. CONCLUSIONS AND FUTURE WORK

In order to improve the delay performance of multislotted messages in a multiple access channel with long propagation delay, a multichannel random access protocol called fast multichannel slotted Aloha (FMCSA) is proposed in this paper. FMCSA combines random access with packet level FEC for new messages and scheduled retransmissions for partially received messages. Through analysis and simulations, we show that FMCSA can achieve higher throughput and lower delay than MCSA. When the system is operating at relatively low load region, the short messages can be delivered in their first attempts with very high probability. We also show that the improved performance of FMCSA compared to MCSA is robust to the FEC code rate, channel bandwidth, terminal population, arrival patterns, slot size as well as message length.

In this paper, we assume all the channels has the same bandwidth and the traffic generated by all the terminals is statistically indistinguishable from each other. We further assume that each terminal is equipped with only one transmitter and all the transmitters send packets with the same power. In the future, we would like to explore the case where the traffic loads at the terminals are heterogeneous, which happens when some terminals are used to connect a small local area network while others are used to connect a single personal computer. It is desirable to assign more bandwidth to those heavier loaded terminals either by increasing the transmission power [12][9] or by allowing them to use more than one transmitters to send packets in parallel. We are also considering to extend FMCSA protocol to a random reservation protocol which is used to transfer not only short messages but also long messages.

## REFERENCES

- [1] N. Abramson. Internet access using VSATs. *IEEE communication magazine*, July 2000.
- [2] Gun Akkor and J. S. Baras. IP multicast via satellite: a survey. Technical report, CSHCN TR 2003-1, <http://www.isr.umd.edu/CSHCN>, January 2003.
- [3] P. Badford and M. Crovella. Generating representative web workloads for network and server performance evaluation. In *ACM SIGMETRICS*, 1998.
- [4] Yitzhak Birk and Yaron Keren. Judicious use of redundant transmissions in multichannel Aloha networks with deadlines. *IEEE J. Select. Areas Comm.*, 17:257–269, February 1999.
- [5] Hyoung-Kee Choi and John O. Limb. A behavior model of web traffic. In *International Conference of Networking Protocol '99*, September 1999.

- [6] M. E. Crovella, M. S. Taqqu, and A. Bestavros. *Heavy-tailed probability distributions in the world wide web, Chapter 1 of a practical guide to heavy tails*. Chapman & Hall, 1998.
- [7] Leonard Kleinrock and Fouad A. Tobagi. Packet switching in radio channels: Part I-Parrier sense multiple access modes and their throughput-delay characteristics. *IEEE Transaction on Communications*, 23(12):1400–1416, December 1975.
- [8] Tho Le-Ngoc and I. Mohammed Jahangir. Performance analysis of CFDAMA-PB protocol for packet satellite communications. *IEEE Transactions on Communications*, 46(9), September 1998.
- [9] W. Luo and A. Ephremides. Power levels and packet lengths in random multiple access. *IEEE Transactions on Information Theory*, 48(1), January 2002.
- [10] Bruce A. Mah. An empirical model of HTTP network traffic. In *IEEE INFOCOM*, 1997.
- [11] A. J. McAuley. Reliable broadband communication using a burst erasure correcting code. In *ACM SIGCOMM'90*, September 1990.
- [12] J. J. Metzner. On improving utilization in ALOHA networks. *IEEE Transactions on Communications*, 24:447–448, April 1976.
- [13] P. D. Mitchell, D. Grace, and T. C. Tozer. Comparative performance of the CFDAMA protocol via satellite with various terminal request strategies. In *IEEE GLOBECOM'01*, pages 2720–2724, 2001.
- [14] Ramaswamy Murali and Brian L. Hughes. Random access with large propagation delay. *IEEE/ACM Transaction on Networking*, 5(6):924–935, December 1997.
- [15] J. Nonnenmacher, E. W. Biersack, and D. Towsley. Parity-based loss recovery for reliable multicast transmission. *IEEE/ACM Transaction on Networking*, 6(4):349–361, August 1998.
- [16] D. Raychaudhuri. ALOHA with multipacket messages and ARQ-type retransmission protocols - throughput analysgs. *IEEE Transaction on Communications*, 32(2):148–154, February 1988.
- [17] D. Raychaudhuri. Selective reject ALOHA/FCFS: an advanced VSAT channel access protocol. *International Journal of Satellite Communications*, 7(3):435–447, 1990.
- [18] L. Rizzo. On the feasibility of software FEC. Technical report, DEIT, <http://info.iet.unipi.it/luigi/fec.html>, January 1997.
- [19] F. D. Smith, F. Hernandez, K. Jeffay, and D. Ott. What TCP/IP protocol headers can tell us about the web. In *ACM SIGMETRICS*, 2001.
- [20] Tak-Shing Yum and Eric W. M. Wong. The scheduled-retransmission multiaccess (SRMA) protocol for packet satellite communication. *IEEE Transactions on Information Theory*, 35(6), November 1989.