

Simulations for Resource-Allocation Protocol Optimization for MIL-STD-188-186 over a UHF SATCOM Network

Edward W. Chandler, *Life Senior Member, IEEE*

Abstract— A simple simulation tool was previously developed to simulate the operation and performance of demand-assigned communication networks having time-varying data traffic patterns. Such networks use protocols to increase or decrease the resources allocated to a network node as the resource needs at that node change. These resource-allocation protocols have parameters such as numerical thresholds for requesting or releasing resources. The additional simulator development and execution described herein allows a comparison of performance metrics that result when using the existing MIL-STD-188-186 protocol versus using a proposed alternative protocol. The network behavior revealed by the simulation results also allows examining network performance trade-offs as values for the protocol parameters are adjusted. Although commercially available simulation tools could be used for these simulations, such tools are often expensive and sometimes complicated to adapt to new and not yet standardized protocols. The simple simulation tool that was previously developed is described in an earlier paper and is programmed as an Excel spreadsheet. It has now been adapted to allow comparing an existing resource-allocation protocol versus a proposed protocol for MIL-STD-188-186. The simulator allows examination of performance metrics such as the average number of assigned time slots per frame assigned to a transmitting node that has a specified message generation rate, the percentage of generated messages that are discarded prior to transmission due to being queued for an excessive time, and a histogram showing the percentage of messages transmitted with each possible message delivery time.

Index Terms—network, optimization, performance, protocol, SATCOM, simulation.

I. INTRODUCTION

A simple simulation tool was developed and is described in an earlier paper [1] with the primary objective of determining the values of communication network performance metrics as different candidate protocols are used and as protocol parameter values are adjusted. Although commercially available simulation tools could be used for these simulations [2]-[5], such tools are often expensive and sometimes complicated

to adapt to new and not yet standardized protocols.

The primary contribution offered by the earlier paper [1] was the Excel-based simulation tool that is relatively easy to program to simulate random message data generation, priority-based message queues, and resource request/release protocols.

The main contribution of this paper is the further development of the simulator to execute the queuing and resource-allocation protocols for MIL-STD-188-186 [6] (both the currently specified protocol and a proposed alternative protocol), to evaluate advantages of the proposed protocol, and to evaluate the simulation results with different parameter values.

Presented here is a description of how this simulation tool was adapted to compare the currently specified MIL-STD-188-186 [6] resource-allocation protocol with a proposed alternative protocol. The simulation tool allows easy comparisons of message-delivery performance results as candidate protocols are tried, and as parameter values within a protocol are adjusted.

MIL-STD-188-186 [6] specifies a set of protocols for a message delivery mechanism that typically operates over an ultra-high frequency (UHF) satellite communication (SATCOM) network for military applications. MIL-STD-188-186 is undergoing a revision to modify a resource-allocation protocol specified within the standard, to result in improved message-delivery performance. This simulation tool has been programmed to execute and evaluate a proposed alternative protocol for this revision and determine protocol parameter values that provide advantageous message-delivery performance.

The simple simulation tool has three sections: a message generator, a message queue, and a resource-allocation protocol processor. The message generator and message queue sections are both described in detail in an earlier paper [1], and a summary description of those simulator sections will be given here. The way in which the message queue was recently further developed to emulate the operation of the MIL-STD-188-186 queue will also be described here.

Descriptions of both the existing MIL-STD-188-186 resource-allocation protocol and the proposed alternative protocol will be given.

Manuscript received January 10, 2019; revised March 5, 2019. Date of publication May 24, 2019. Date of current version June 3, 2019.

E. W. Chandler is with Milwaukee School of Engineering, Milwaukee, WI 53202 USA (e-mail: chandler@msoe.edu).

Digital Object Identifier (DOI): 10.24138/jcomss.v15i2.689

The manner in which a resource-allocation protocol is coded into the simulator will be illustrated. Some of the simulation results will be given, showing different message-delivery performance results between the two protocols, and showing different levels of performance as protocol parameter values are changed.

It is assumed that data messages are transmitted using RF burst transmissions that each occur on an RF channel and each occur within a defined time slot. On each available RF channel, time division multiple access (TDMA) is used. With TDMA, a fixed length of time called a time frame contains multiple time slots of different lengths. Each time slot can be assigned to a particular transmitting node for its exclusive use (in which case it is called an assigned time slot) or can be designated as a random-access time slot. A random-access time slot can be used by any transmitting node on a contention basis if the node has a sufficiently high priority message to transmit and if that message cannot be accommodated by an assigned time slot. For the system being simulated here, the time frame has a length of 1.38667 seconds and each time slot intended for message data transmissions will have one of two possible sizes (or lengths in units of time), a large time slot or small time slot.

Each node that transmits messages has one or more assigned time slots permanently assigned to it and can request from a controller the assignment or de-assignment of additional time slots based on need. Each node that transmits messages exercises a resource-allocation protocol that (a) monitors either the size of its queue of untransmitted messages or the rate at which messages are being generated, (b) requests additional resources (that is, an additional time slot) if what it is monitoring exceeds some threshold, and (c) releases excess time slots if what it is monitoring falls below some threshold.

In a typical application, a commonly used message length is used to determine the size of the large time slots used by the system. A message longer than the typical message length would then require multiple time slots for its transmission. The commonly used message length used to determine the time slot size (as an example, a 224-byte message) is defined as having a length of one block; a message having a length of 616 bytes in this example would then have a length of 2.75 blocks. The small time slot used in a typical system is sized to accommodate a much smaller message (as an example, a 32-byte message, which would have a length of $32/224$ block or 0.1429 block). The small time slot can accommodate the transmission of a short message or a portion of a larger message. For example, a small time slot that accommodates 32 bytes could be used to transmit one-seventh of a 224-byte (or one-block) message.

II. MESSAGE GENERATOR

The simple simulation tool described herein is implemented as an Excel spreadsheet. For a large portion of the spreadsheet, each row represents the next short time interval that follows the preceding time interval that is represented by the spreadsheet row just above it. Each short time interval is referred to as a frame, and represents a TDMA frame (see section I). Several columns within the spreadsheet are used to implement the message generator, with one column indicating the total number

of messages generated in each frame designated by row. In the application of interest, messages each have a priority level, being one of four priorities, and four columns are used to indicate the number of messages generated at each of the four priority levels, within each frame designated by row. As an example, the message generator results shown in Table I show the number of messages at each of four priorities generated each succeeding frame, when the Excel Data Analysis tool called *Random Number Generator* is used to generate Poisson distributed messages with an average rate of three messages per frame during non-surge conditions and a rate of 10 messages per frame during a surge. In the example shown in Table I, each of the four priorities is equally likely during non-surge conditions but the random selection of priority becomes 80% priority-1 and 20% priority-2 during a surge. In Table I, there is a surge between frame 8 and frame 11, inclusive.

TABLE I
EXAMPLE OF POISSON-DISTRIBUTED MESSAGE GENERATION

FRAME NUMBER	NUMBER OF MESSAGES GENERATED				Total
	at Pri-1	at Pri-2	at Pri-3	at Pri-4	
0	2	0	1	0	3
1	1	0	1	1	3
2	0	2	0	0	2
3	0	2	1	1	4
4	3	2	2	1	8
5	0	3	3	1	7
6	0	1	2	0	3
7	1	0	1	0	2
8	6	0	0	0	6
9	6	2	0	0	8
10	6	4	0	0	10
11	10	1	0	0	11
12	2	0	1	0	3
13	0	1	1	0	2
14	0	0	1	1	2
15	0	0	0	0	0
16	3	0	0	0	3
17	0	0	0	2	2

Additional columns have been programmed to handle other message generator features. For example, any of $N_{LENGTHS}$ possible message lengths could be accommodated for the generated messages, allowing randomly distributed message lengths. For each priority level there could be $N_{LENGTHS}$ columns that each show the number of generated messages having a particular length for that priority level. In the simulations presented below, all messages have the same length, so $N_{LENGTHS} = 1$.

III. MESSAGE QUEUE

Data transmissions are expected to be processed by a message queue within each transmitting network node, with highest priority messages transmitted first, followed by lower priorities. At each priority level, messages are expected to be removed from the queue and transmitted on a first-in-first-out (FIFO) basis but a message is removed from the queue and discarded if its time in queue exceeds a predefined threshold. It is assumed that any messages newly generated in a particular

frame are first available for transmission in the following frame.

In each frame i , the capacity $C(i)$ represents the number of blocks (defined in section I) that the network node can transmit in that frame. For example, if $C(3) = 2.7$ blocks, this means that in frame number 3 the resources allocated to the node provide a capacity of 2.7 blocks. It is generally assumed that messages can have different lengths. If at the beginning of frame number 3 the network node had in its queue three messages at priority-1, with the first (that is, oldest) having a length of 1 block, the second having a length of 3 blocks, and the third having a length of 0.85 block, then in that frame the node would use its capacity, $C(3) = 2.7$ blocks, to transmit all of the first message and 1.7 blocks (of the 3 blocks) for the second message. It would then retain the untransmitted portion of the second message (having length 1.3 blocks) followed by the message having length 0.85 block. At the end of frame number 3, it would have in its priority-1 queue two messages, one with remaining length 1.3 blocks and the other with length 0.85 block, followed by any new priority-1 messages that were generated during frame number 3.

In each frame, the message queue uses its capacity to transmit as much of its priority-1 data as possible, in FIFO order, and then transfers the remaining capacity (if any) to the priority-2 queue for scheduling transmissions in a similar way. The priority-2 queue, after transmitting as much of the queued priority-2 data as possible, in FIFO order, transfers any remaining capacity to the priority-3 queue, etc.

The message queue at each priority level keeps track of how old the remaining data segments are for each message within its queue. Any message data segment that is not transmitted in a particular frame i due to insufficient capacity, if such data had an age of j frames during frame number i , would be marked as having an age of $j+1$ frames in the spreadsheet row that shows the processing for frame $i+1$. As a simple example, assume that the network node begins at frame number 1 with a capacity of 1 block each frame, and assume that all generated messages are each of length 1 block. Assume that in frames numbered 0, 1, 2, 3, 4, and 5, the number of messages generated each frame, each having length 1 block, is 2, 3, 0, 1, 3, and 0. Then in each frame, the queue will have stored the number of messages of each age as indicated in Table II. In each frame, the messages generated in that frame are all stored as messages having an age of 0 as of the end of that frame and are first available for transmission in the next frame.

In Table II, of the two messages generated in Frame Number 0, one gets transmitted in Frame Number 1 (because there is only one time slot) and the other is shown at the end of Frame Number 1 to have an age of 1 frame old. In Frame Number 2, the second message generated in Frame Number 0 gets transmitted in the one time slot available, and the three messages generated in Frame Number 1 are all queued at the end of Frame Number 2 with an age of 1 frame old.

Beginning in Frame Number 5, with a capacity of two messages per frame, the one message that was 3 frames old at the end of Frame Number 4 and the one message that was 1 frame old at the end of Frame Number 4 get transmitted in those two time slots in Frame Number 5, and the three messages that

were generated in Frame Number 4 continue to be queued in Frame Number 5 as 1-frame-old messages at the end of Frame Number 5.

TABLE II

EXAMPLE OF PRIORITY-1 MESSAGE QUEUE, ASSUMING THAT TRANSMISSION CAPACITY IS ONE MESSAGE/FRACTION IN FRAMES 1 – 4, TWO MESSAGES/FRACTION IN FRAMES 5 – 8, AND THREE MESSAGES/FRACTION IN FRAMES 9 – 11.

PRIORITY-1 MESSAGES IN QUEUE AT PRODUCER BEING SIMULATED (IN FIFO ORDER)						
FRAME NO.	NUMBER OF PRIORITY-1 MESSAGES GENERATED THIS FRAME	NUMBER OF PRIORITY-1 MESSAGES QUEUED AT THE END OF THIS FRAME, HAVING AGE (IN FRAMES):				
		Age = 0 frame	Age = 1 frame	Age = 2 frames	Age = 3 frames	Age = 4 frames
0	2	2	N/A	N/A	N/A	N/A
1	3	3	1	0	0	0
2	0	0	3	0	0	0
3	1	1	0	2	0	0
4	3	3	1	0	1	0
5	0	0	3	0	0	0
6	0	0	0	1	0	0
7	1	1	0	0	0	0
8	6	6	0	0	0	0
9	6	6	3	0	0	0
10	10	10	6	0	0	0
11	11	11	10	3	0	0

The type of manual checking of queue operation as just described was done for several of the simulation results observed and was the primary method of validating the operation of the simulator. This is because no other simulation tool was known to be capable of easily simulating the proposed resource-allocation protocol, and neither the existing MIL-STD-188-186 protocol nor the proposed protocol had yet been implemented within equipment that could be tested.

In each frame, at each priority level, the oldest message in the queue is scheduled first for transmission, followed by the next oldest, etc. Any untransmitted message(s) during a particular frame are recorded as having gained 1 frame in age when considered in the next frame. Any messages that cannot be transmitted before they reach a programmable expiration age are marked as having been deleted from the queue due to the queue time exceeding a specified threshold.

A table within the message queue section of the simulation spreadsheet shows summary message-delivery performance data for that priority level, which is shown by example on Fig. 1. The Priority-1 performance results shown on Fig. 1 were obtained by configuring the simulator to generate Poisson distributed messages, all at priority 1, at an average rate of 1.5 messages per frame, using an Excel random-generator seed of 79, operating with the proposed protocol having parameter values (described later) of $X_1=X_2=0.95$, $X_3=1.25$, $X_4=1.3$, and $S=20$. As shown on Fig. 1, among the 754 messages generated over the 500-frame simulation interval, six were discarded due to exceeding the maximum queue time, one was still queued at the end of the simulation interval, and the other 747 were transmitted within the allowed 7-frame maximum queue time.

IV. RESOURCE-ALLOCATION PROTOCOL

For a demand-assigned type of network, protocols referred to as resource-allocation protocols are used by each network node to determine when additional resource allocation, sometimes called bandwidth, needs to be requested and assigned. Protocols could determine this need based either on growing queue sizes for data traffic needing to be transferred or on observations of increased data traffic generation rates, the latter of which is used by the protocols simulated in this study. These protocols also

determine the conditions for releasing resources based on observations of an unacceptably high fraction of allocated resources not being used.

Examples of protocol parameters that can be adjusted to optimize performance metrics include (a) *S*, the observation time window or span over which data traffic and assigned resource utilizations are measured and compared to thresholds to determine when additional resources should be requested or released, and (b) the threshold values that must be met to trigger a request for more resources or a release of resources.

This component of the simulation tool executes the protocol for increasing and reclaiming resources, and allows the setting of protocol parameter values. The format for the resource-allocation protocol section of the simulator, when emulating the proposed replacement protocol for MIL-STD-188-186 is as described in the next section and is depicted on Fig. 2. Although partitioned to a different cell area of the Excel spreadsheet, when this simulation component is used, it operates in a manner that becomes integrated with the message queue component of the simulator.

If a request to either add or release resources is sent by the network node, then the simulator must be programmed to use an appropriate time required for the request to be transferred and time required for that request to be acted upon. These time delays can be either random or deterministic. They are programmed into the simulator so that capacity changes become effective with appropriate delays. For the simulations presented herein, the delay time was set to be a deterministic constant and was set to equal three frame times as indicated on Fig. 2. As an example shown on Fig. 2, it can be seen that a request for more resources (shown as REQUEST MORE BW FLAG, where BW is an abbreviation for bandwidth) was determined to be needed at the start of Frame No. 3, it is then assumed that the request is sent during Frame No. 3, and the Granted Change indeed occurred and became effective in Frame No. 6. After a capacity change becomes effective, the new capacity constitutes the resources that are then available for use by the network node, and the network node protocols must immediately use the new capacity in its determinations of when to thereafter request to either add or release resources.

PERFORMANCE STATISTICS: (Priority-1)	
Number of frames for simulation:	
Nframes=	500 frames
Number of Priority-1 messages generated:	
Npri-1-msgs=	754 messages
Number of Pri-1 messages discarded (exceeded max queue time, RQT):	
Npri-1-RQT=	6 reports
Note: RQT =	7 frames
Number of Pri-1 messages queued for transmission at end of simulation:	
Npri-1-queued=	1 reports
Number of Pri-1 messages transmitted:	
after 1 frame:	Npri-1-1fr= 416 messages
after 2 frames:	Npri-1-2fr= 198 messages
after 3 frames:	Npri-1-3fr= 63 messages
after 4 frames:	Npri-1-4fr= 16 messages
after 5 frames:	Npri-1-5fr= 5 messages
after 6 frames:	Npri-1-6fr= 32 messages
after 7 frames:	Npri-1-7fr= 17 messages
after 8 frames:	Npri-1-8fr= 0 messages
after 9 frames:	Npri-1-9fr= 0 messages
after 10 frames:	Npri-1-10fr= 0 messages
after 11 frames:	Npri-1-11fr= 0 messages
after 12 frames:	Npri-1-12fr= 0 messages
after 13 frames:	Npri-1-13fr= 0 messages
after 14 frames:	Npri-1-14fr= 0 messages
TOTAL TRANSMITTED:	747 messages
AVERAGE NO. ASSIGNED TIME SLOTS PER FRAME:	
	2.10686 slots/frame

Fig. 1. Example priority-1 performance data.

Optional Manual Override of Settings		Report Generation Rates:				RESOURCE-ALLOCATION PROTOCOL (RAP) SECTION (This section implements the ARP Mar-2018 resource-allocation protocol.)															
X1= 0.8		Non-surge: 1 msgs/frame				Surge: 1.5 msgs/frame				There are 0 surges, with first one in frame.											
X2= 1		The values in columns GX & GY are determined at the start of each frame.				The values in these 4 columns just below are determined at the start of each frame.				Determined at start of each frame:											
X3= 1.1																					
X4= 1.25																					
S= 20																					
Sextra= 87																					
See GZ123																					
/-- At start of frame --/		/--- This frame ---/				Rqst more for Pri-1-2		Rqst more for Pri-1-4		Release for Pri-1-2		Release for Pri-1-4		Ceiling of (209* X3*M*1)		REQUEST MORE BW FLAG		RELEASE BW FLAG		Ceiling of (209*X4* Mextra*1)	
Frame No.	Capacity	Mrpts12	Mrpts1234	X1*C*M	X2*C*87	BW Flag	BW Flag	BW Flag	BW Flag	C*M - (209* X3*M*1)	Ceiling	FLAG	FLAG	FLAG	FLAG	FLAG	FLAG	FLAG	FLAG	FLAG	FLAG
1	0.14286	2	2	2.285714286	12.42857143	0	0	0	0	22	-19.143	0	0	0	0	0	0	0	0	0	108.7511962
2	0.14286	2	2	2.285714286	12.42857143	0	0	0	0	22	-19.143	0	0	0	0	0	0	0	0	0	108.7511962
3	0.14286	3	3	2.285714286	12.42857143	1	0	0	0	22	-19.143	1	0	0	0	0	0	0	0	0	108.7511962
4	0.14286	4	4	2.285714286	12.42857143	1	0	0	0	22	-19.143	1	0	0	0	0	0	0	0	0	108.7511962
5	0.14286	4	4	2.285714286	12.42857143	1	0	0	0	22	-19.143	1	0	0	0	0	0	0	0	0	108.7511962
6	1.14286	9	9	18.28571429	99.42857143	0	0	0	0	22	0.85714	0	0	0	0	0	0	0	0	0	108.7511962
7	1.14286	9	9	18.28571429	99.42857143	0	0	0	0	22	0.85714	0	0	0	0	0	0	0	0	0	108.7511962
8	1.14286	10	10	18.28571429	99.42857143	0	0	0	0	22	0.85714	0	0	0	0	0	0	0	0	0	108.7511962
9	1.14286	13	13	18.28571429	99.42857143	0	0	0	0	22	0.85714	0	0	0	0	0	0	0	0	0	108.7511962
10	1.14286	13	13	18.28571429	99.42857143	0	0	0	0	22	0.85714	0	0	0	0	0	0	0	0	0	108.7511962
11	1.14286	13	13	18.28571429	99.42857143	0	0	0	0	22	0.85714	0	0	0	0	0	0	0	0	0	108.7511962
12	1.14286	14	14	18.28571429	99.42857143	0	0	0	0	22	0.85714	0	0	0	0	0	0	0	0	0	108.7511962
13	1.14286	17	17	18.28571429	99.42857143	0	0	0	0	22	0.85714	0	0	0	0	0	0	0	0	0	108.7511962

Fig. 2(a). Example Resource Allocation Protocol (RAP) simulator implementation for proposed replacement algorithm for MIL-STD-188-186 (part 1).

	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX
	It is assumed just below that a request is granted (and is effective) 3 frames after the frame in which the request is sent. Note that this could be modified to allow simulator user to enter the number of frames that it takes to effect a request (e.g., 3, 4, or 5).			
(1/209)*	No. frames	No. frames	Granted	
Ceiling of	Rqst More	Rqst Rels	Change	
(209*X4*	is Pending	is Pending	This Frame	
Mextra*1)				
108.7511962	0	0	0	
108.7511962	0	0	0	
108.7511962	0	0	0	
108.7511962	1	0	0	
108.7511962	2	0	0	
108.7511962	3	0	1	
108.7511962	0	0	0	
108.7511962	0	0	0	
108.7511962	0	0	0	
108.7511962	0	0	0	
108.7511962	0	0	0	
108.7511962	0	0	0	
108.7511962	0	0	0	
108.7511962	0	0	0	

Fig. 2(b). Example Resource Allocation Protocol (RAP) simulator implementation for proposed replacement algorithm for MIL-STD-188-186 (part 2).

V. MIL-STD-188-186 RESOURCE-ALLOCATION PROTOCOLS

The current version of MIL-STD-188-186 [6] specifies, in its Appendix A, a resource-allocation protocol that is based on a single criterion. The protocol has the network node request an additional time slot when

$$M \geq 85(K-1) - 80, \quad (1)$$

where M is the total generated message data, in blocks (defined in section I), generated over the past 87 frames, and K is the number of assigned time slots (defined in section I) that the node currently has as allocated resources. The same section of that MIL-STD specifies that the network node is expected to request the release of an assigned time slot if K is greater than 1 and

$$M \leq 85(K-1) - 50. \quad (2)$$

One of the problems with this resource-allocation protocol currently specified in MIL-STD-188-186 is that decisions are based on the amount of generated message data over 87 frames, which is a span of 120.6 seconds, so the protocol will either react rather slowly to surges in message data generation or, with the use of a relatively small threshold for deciding to request an additional time slot, can react sufficiently fast but will typically then result in having more added time slots than what is really needed in the steady state. In order to accommodate surges in higher priority messages (at priorities 1 and 2) that typically have short maximum queue times, the -80 term within the threshold $85(K-1) - 80$ produces a low threshold for fast reaction but, as the simulations will show, does result in an excessive (that is, wasteful) number of additional time slots being assigned in steady state operation. This motivated the examination of an alternative proposed protocol that separately monitors both high-priority volume and total generated message volume.

Because it was not immediately clear how large should be the value of S , the moving-window time interval over which higher-priority generated messages are counted, the value for S was also left as a parameter with which to experiment through observed simulations.

The proposed alternative protocol for MIL-STD-188-186 resource allocation, described just below, has been simulated and compared with the performance of the existing algorithm, and has been found to result in more efficient resource allocations due to more effective releases of additional time slots when they are no longer needed.

The proposed resource-allocation protocol has the network node separately track the parameters $M_{bytes,1234}$ and $M_{bytes,12}$, which are defined in Table III. The parameters S and X_1 through X_4 within the resource-allocation protocol are also defined in Table III, and simulations that experiment with various values for these five parameters are used to determine values that result in preferred network performance, that is, values that result in more desirable network operation. In some cases, adjusting the value of one of these parameters will cause one network performance metric to become better while causing a different metric to become worse.

TABLE III
PARAMETERS PERTAINING TO PROPOSED RESOURCE-ALLOCATION PROCESS

Parameter	Definition
$M_{bytes,1234}$	the total generated message data, in bytes, generated over the past 87 frames
$M_{bytes,12}$	the total generated message data, in bytes, generated over the past S frames, for only priority-1 and priority-2 messages
S	the sliding-window span interval used for the tracking of priority-1 and priority-2 messages
C	the transmit capacity of all assigned time slots that the node currently has as allocated resources, in bytes
C_{last}	the transmit capacity of the time slot that was most recently assigned to the network node as an additional resource, in bytes
X_1	a threshold parameter used when determining whether an additional time slot is needed, based on generated message data at priorities 1 and 2
X_2	a threshold parameter used when determining whether an additional time slot is needed, based on generated message data at all priorities
X_3	a threshold parameter used when determining whether an assigned time slot can be released, based on generated message data at priorities 1 and 2
X_4	a threshold parameter used when determining whether an assigned time slot can be released, based on generated message data at all priorities

The proposed protocol has a Rule #1 that has the network node *request an additional time slot whenever either of the two following conditions are met:*

- $M_{bytes,1234}$ exceeds $(87X_2C)$, where each of these parameters is defined in Table III, or

- (b) $M_{bytes,12}$ exceeds (SX_1C) , where again each of these parameters is defined in Table III.

Note that the transmit capacity called C in the above described protocol rule is the same as the parameter called $C(i)$ in section III above. The frame number i that is referred to in section III will, for the above protocol rule, always be equal to the frame number for the current frame during which the rule is being exercised.

The proposed protocol has a Rule #2 that has the network node *request the release of an assigned time slot whenever both of the two following conditions are met:*

- (a) $M_{bytes,1234}$ falls below $87C - \text{ceiling}(87X_4C_{last})$, where each of these parameters is defined in Table III, and
 (b) $M_{bytes,12}$ falls below $SC - \text{ceiling}(SX_3C_{last})$, where again each of these parameters is defined in Table III.

Note that, regarding the C_{last} parameter in the above rule, which is defined in Table III, a transmitting node would need to keep track of the transmit capacity of each added time slot as additional time slots are added, in order to use the correct current value of C_{last} as time slots are released. Some systems operate with the use of only one size of time slot for those that are added and released, in which case the value of C_{last} would be a constant and would not need to be separately tracked for each time slot added or released.

It should also be noted that if all messages have the same length, which is the case for the simulations presented here, and if only two time slot sizes are used, the larger of which accommodates a message and the smaller of which accommodates one-seventh of a message, then the proposed resource-allocation protocol can track generated message data in units of reports instead of units of bytes, that is, track $M_{rpts,1234}$ and $M_{rpts,12}$, instead of tracking $M_{bytes,1234}$ and $M_{bytes,12}$, as indicated by columns shown on Fig. 2. The above-stated proposed protocol Rule #1 and Rule #2 would then be modified by replacing $M_{bytes,1234}$ with $M_{rpts,1234}$, replacing $M_{bytes,12}$ with $M_{rpts,12}$, defining C as the transmit capacity of all assigned time slots that the node currently has as allocated resources, in reports, and using a constant value of C_{last} equal to one report. For the simulations presented herein, this is indeed how the simulator was coded.

VI. EXAMPLE SIMULATION RESULTS

The simulator results show that the proposed protocol gives much better performance metrics than the existing MIL-STD-188-186 protocol. Using either the existing MIL-STD-188-186 protocol or the proposed protocol, it is found that a very high percentage of high-priority messages are transmitted prior to an established report expiration time, but this is possible using the existing MIL-STD-188-186 protocol only because its threshold for requesting additional time slots is set very low and this results in very excessive over-assignment of additional time slots and failure to release all of them that are no longer needed.

To compare the behaviors of the existing MIL-STD-188-186 and proposed resource-allocation protocols described in section V, the simulator was first programmed to execute each of these two protocols with a transmitting node that generated only

priority-1 messages with message statistics as described in the paragraph just before section IV. The currently specified protocol in MIL-STD-188-186 quickly requested additional time slots resulting in three additional time slots being assigned by frame number 14, and a fourth additional time slot assigned in frame number 62; however, none of these four additional time slots was ever released during the 500-frame simulation. The proposed protocol, using parameter values $X_1=0.97$, $X_2=1$, $X_3=1.1$, $X_4=1.05$, and $S=14$, quickly requested additional time slots resulting in two additional time slots being assigned by frame number 18, and first requested a third additional time slot near the end of the 500-frame simulation (resulting in the third additional time slot being granted in frame number 500).

Over the span of the 500-frame simulation, the proposed protocol on the average resulted in 1.95 additional time slots per frame, whereas the existing MIL-STD-188-186 protocol resulted in an average of 3.83 additional time slots per frame, nearly double that for the proposed protocol. This was due to the existing protocol not releasing unneeded resources. The proposed protocol resulted in 6 of the generated 754 messages (which is just under one percent) being discarded due to being queued for a time that exceeded the maximum queue time, whereas the existing MIL-STD-188-186 protocol resulted in no discarded messages. For the proposed protocol, it was found that lowering the X_1 parameter value would reduce the number of discarded messages but would increase the average number of additional time slots per frame, over the 500-frame simulation interval. For example, with $X_1=0.75$, the number of discarded messages went down from 6 to 4 but the average number of additional time slots went up to 2.15.

As a second example of simulation results used to optimize protocol parameter values, the simulation was executed using Poisson distributed message arrivals with only one average message generation rate, that is, no surge conditions. The initially used X_2 and X_4 parameter values were 1.0 and 1.25, respectively. Simulations using these parameter values revealed that, as shown in Table IV, with $X_4 = 1.25$, if message-generation conditions cause resources to be added due to an occasional random time interval that by chance had more message data to send than the typical amount of data over that interval, then the network node would very likely *not* release these added resources after the momentary interval with higher volume of data.

TABLE IV
RESOURCE-RELEASE PROBLEM WITH NONIDEAL X_4 PARAMETER VALUE

	$X_2 = 1$ $X_4 = 1.25$	$X_2 = 0.95$ $X_4 = 1.05$	$X_2 = 0.9$ $X_4 = 1.1$
Frame numbers when extra resources assigned	36-800 (resources not released)	36-49	36-90
Average no. of slots/frame assigned over 800-frame simulation	1.099 (excessive resource usage)	0.1603	0.2116

The simulation revealed that an X_4 value of 1.25, when used with an X_2 value relatively close to 1.0 (such as 0.95), resulted in a node *not* requesting additional resources when the data generation was less than 95% of capacity (that is, when the data generation rate resulted in at least 5% of resources wasted), but also allowed 25% of resource wastage before releasing resources. By changing the X_4 parameter value to 1.05, the simulations showed that the excess resources would indeed be released after the momentary period when they were needed. The simulation also showed that with appropriate X_2 and X_4 parameter values, the resource assignment/usage over the 800-frame simulation was 0.1603 slots per frame instead of 1.099 slots per frame, which represents a huge reduction in resources used. It should be noted that with any of the parameter-value choices given in Table IV, there were no high-priority messages discarded due to being queued too long. Another observation from the simulations was that if the parameter values are not appropriately chosen then oscillations can occur as continuous cycles of resource assignment followed by resource release. For example, in the simulations having results shown in Table IV, if the X_2 and X_4 parameter values were chosen as 0.3 and 1.1, respectively, then there was a total of twelve time intervals (not just one) that had extra resources assigned and subsequently released.

As a third example of parameter optimization, simulations were run with periods of higher message-generation-rate surges. The message generation rates, 0.13 messages per frame during non-surge and 1.65 messages per frame during each surge, were believed to be representative of typical system operation. First the X_1 and X_3 parameters were adjusted to each of several values, with message-delivery and resource-assignment performances observed in each case. Additional simulations were run using alternative values for the span parameter called S . Some of these results are shown in Table V.

It is clear from Table V that a somewhat larger value for S (in this case 20) resulted in far fewer changes in assignments, with little effect on other performances. For the Table V simulations, none of the priority-1 messages were discarded due to excessive queue times, and all three of the results shown in Table V had the same number of priority-2 messages discarded due to excessive queue times.

As a fourth example, for a representative time segment during which 61 priority-1 messages were generated, the simulator was used to determine the number of messages that were delayed beyond a maximum queue time and therefore discarded, called N_{MXQT} , and the number of messages that were delayed to at least half that value (but were subsequently sent before exceeding the maximum queue time), called $N_{0.5MXQT}$. The results are given in Table VI below.

As can be seen in Table VI, for the span parameter S , a value no greater than 12 resulted in no priority-1 messages needing to be discarded over the duration of the simulation. However, per Table V, the somewhat larger S parameter values tended to produce fewer requests for increasing and decreasing resources allocated.

TABLE V
SIMULATION RESULTS WITH VARIOUS X_1 , X_3 , AND S PARAMETER VALUES

	$X_1 = 1$ $X_3 = 1.25$ $S = 14$	$X_1 = 0.95$ $X_3 = 1.05$ $S = 14$	$X_1 = 0.9$ $X_3 = 1.1$ $S = 20$
Number of requested additions and releases of resources	23	20	13
Maximum number of added resources at any time	3 (occurred in 5 intervals)	3 (occurred in 2 intervals)	2 (occurred in 7 intervals)
Average no. of slots/frame assigned over 800-frame simulation	1.0591	1.0166	0.9991
Average priority-1 message delivery time (in frames)	1.210	1.235	1.296

TABLE VI
EXAMPLE OPTIMIZATION OF SPAN PARAMETER, S

Span, S	No. of Messages Discarded, N_{MXQT}	No. of Messages with Significant Delay, $N_{0.5MXQT}$
6	0	14
8	0	18
10	0	16
11	0	13
12	0	11
13	1	10
14	2	7
15	3	7
20	7	5

VII. CONCLUSION

The simulator results showed that the proposed resource-allocation protocol offered a significant improvement in the minimization of wasted resources relative to the results when using the existing MIL-STD-188-186 protocol. The simulation tool also was demonstrated to be an effective tool for determining the effects of adjusting parameters such as threshold values used by the resource-allocation protocol, and is therefore useful in the evaluation of trade-offs such as

lowering the percentage of reports discarded due to being in queue for a time that exceeds an established maximum, at the expense of increasing the average number of additional time slots (or resources) that must be allocated in order to achieve the lowered fraction of messages discarded. In this study, it was determined that it would likely require significant effort to adapt a commercial simulation tool to simulate the resource-allocation protocols evaluated in this study. However, an area of possible future work would be to identify a commercial simulation tool and undertake the necessary adaptations to allow the tool to do such simulations for comparisons with the results from this study.

REFERENCES

- [1] E. W. Chandler, A. Pappas, "Computational Modeling and Simulation of Dynamic Communication Network Resource Allocations in Excel," presented at SpliTech, Split, Croatia, 2018. [Online.] Available: <https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=chandler%20pappas%20dynamic>
- [2] W. T. Kasch, J. R. Ward, J. Andrusenko, "Wireless network modeling and simulation tools for designers and developers", *IEEE Communications Magazine*, vol. 47, issue 3, pp. 120-127, March 2009. <http://dx.doi.org/10.1109/MCOM.2009.4804397>
- [3] A. Kumar, S. K. Kaushik, R. Sharma, "Simulators for Wireless Networks: A Comparative Study", *IEEE International Conference on Computing Sciences (ICCS)*, pp.338-342, 2012., <http://dx.doi.org/10.1109/ICCS.2012.>
- [4] D. M. Nicol, "Utility Analysis of network simulators," *International Journal of Simulation Systems, Science, and Technology*, vol. 4, nos. 3-4, Sept. 2003, ISSN 1473-804X
- [5] J. Duffy, "Simulation tools are as varied as the LANs they model," *Network World*, vol. 15, no. 8, pp. 48-49, Feb. 1998.
- [6] MIL-STD-188-186, 24 August 2009, Department of Defense Interface Standard, "Interoperable Standard for UHF SATCOM Short-Delay Report-Broadcast Service," U.S. DoD.



Edward W. Chandler (M'76-SM'81) was born in Milwaukee, WI, USA in 1953. He received the B.S. degree in electrical engineering from the University of Wisconsin, Milwaukee, in 1975, the M.S. degree in electrical engineering from Illinois Institute of Technology, Chicago, IL, in 1978, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1985. From 1976 to 1977, he was an Electronic Engineer with Motorola, from 1984 to 1986 he was an Assistant Professor at Marquette University, and from 1986 to 1992 he was an engineer with M/A-COM Government Systems (acquired in 1990 by the Titan Corporation). Between 1977 and 1984, and since 1992, he has been an electrical engineering faculty member at MSOE in Milwaukee, WI, USA, where he currently serves as Adjunct Professor. His areas of interest include modulation and coding, wireless network protocols, and network performance simulations. Dr. Chandler is a member of AFCEA and ASEE. He was a recipient of the MSOE Karl O. Werwath Engineering Research Award, the Titan Corporation Noel Amherd Award, and the Marquette University College of Engineering Outstanding Teacher Award.