SCC #53 APPOVED 23 May 2000
Enhancements to RE-NAD

MIL-STD-188-220B: 20 January 1998

5.3.2.1 Scheduler. When the net access is embedded in the radio, a scheduler may be implemented in the DTE or communications processor to organize radio access throughout the network. The scheduler is used to provide a random distribution of timing for channel requests. When a station has data to transmit, it shall calculate the scheduler timer as indicated in Appendix C (C4.4.4.1). When this timer expires, the link layer shall first determine that the previous frame concatenation was transmitted by the physical layer. If the frame concatenation was not transmitted, the link layer shall request its transmission. If a higher precedence individual frame becomes available for transmission, the concatenated frames shall be re-built to include the higher precedence frame. If the previous frame concatenation was transmitted, the link layer shall build a new frame concatenation. This frame concatenation shall then be passed to the physical layer for transmission. Both randomized and immediate scheduler modes are specified in Appendix C (C4.4.4.1.1 ~~and~~ ; C4.4.4.1.5 and C4.4.4.1.6, respectively).

C.4.4.4. ~~Radio embedded~~Radio embedded network access delay (~~RE~~RE~~-~~NAD).

C.4.4.4.1. ~~RE-~~RE-NAD media access. The ~~radio embedded~~radio embedded network access delay (~~RE-~~RE-NAD) DTE data link layer uses a ~~1-persistent~~ channel access protocol between the DTE (DMTD or C.4I system) and DCE which is influenced by radio net voice activity. When the continuous scheduler interval timer (Tc) expires and the previous series of concatenated frames was successfully transmitted, a new series of frames is sent to the physical layer. If there is a pending series of concatenated frames, its transmission is requested again. It should be noted that the physical layer holds the series of concatenated frames when channel access has been denied. If channel access was denied a new Tc timer is calculated and channel access for transmission of the pending series of concatenated frames is requested when the new Tc timer expires. If a higher precedence individual frame becomes available for transmission, the concatenated frames shall be re-built to include the higher precedence frame. For the Type ~~1 acknowledgment~~3, the ~~RE-~~RE-NAD portions in both DTE and DCE are suspended and channel access is controlled by the RHD and TP processes. The ~~RE-~~RE-NAD algorithm is resumed following expiration of the TP timer.

C.4.4.4.1.1. Random schedule interval. In order to achieve high performance radio network communication, efficient channel access and multi-level precedence, a ~~1-persistence~~ fast attack slow decay ~~RE~~RE-~~-~~NAD algorithm is implemented in the DTE. This algorithm uses the "continuous scheduler" concept to provide a random distribution of timing for channel access via the Tc interval timer. The Tc interval timer is the sum of a ~~fixed interval~~voice component and a ~~random interval~~data component. ~~Toffset is the fixed interval. The fixed interval is dependent on the local station's recent use of the channel and~~ The voice component is a fast attack slow decay function permitting voice to immediately slow down the scheduler (fast attack) while gradually speeding the scheduler back up to normal as long as there isn't any voice on the radio net (slow decay). It is described more fully in C.4.4.4.1.2. ~~The random interval is dependent on network population, network connectivity, traffic load and the local station's recent use of the channel.~~The data component is a randomization around net size and the Fload algorithm. It is discussed in C.4.4.4.1.3 and C.4.4.4.1.4.

The value of Tc is recalculated ~~periodically~~ after every local transmission attempt from the expression:

Tc = ~~Toffset~~ voice factor + uniform_random(~~schedint~~SchedulerInterval)

where uniform_random (~~schedint~~SchedulerInterval) is a uniform random number function using the range

0~~-~~ - ~~schedint~~SchedulerInterval.

Uniform_random returns an integer value.

~~The record of transmit traffic load at the local station is updated after every transmission attempt by the local station, which modifies the value of Toffset. Thus, TC is to be updated after every local transmission attempt.~~

C.4.4.4.1.2. Voice component. The voice factor in C.4.4.4.1.1 is a value bound by its minimum and maximum limits. The maximum voice factor (range= 0.3 to 10.0 sec) is the upper bound on the value of the voice factor. The minimum voice factor (range=0.3 to 10.0 sec) is the lower bound on the value of the voice factor. The initial voice factor value shall be the minimum voice factor value. Detection of voice on the radio net increments the voice factor value while the absence of voice decrements the voice factor value.

C.4.4.4.1.2.1 Fast attack. Voice detection shall increment the voice factor by the voice factor increment value (range=0.0 sec to 10.0 sec) as indicated below.

a. If the voice factor is at the minimum voice factor value (see C.4.4.4.1.2 above), the scheduler is incremented immediately to protect the next voice hit.

b. Otherwise, the increment occurs at the next scheduler expiration

The voice factor value is upper bounded by the maximum voice factor (see C.4.4.4.1.2 above)

C.4.4.4.1.2.2 Slow decay. The voice factor shall be decremented every time the NAD expires by the voice decrement value (range=0.0 sec to 10.0 sec). The voice factor value is lower bounded by the minimum voice factor. ~~C.4.4.4.1.2. Calculation of the scheduler offset. The parameter "Toffset" in C.4.4.4.1.1 is a function of the average transmit duration. A transmission is composed of unnumbered, supervisory and information frames. Combinations of these frames are concatenated into a series of frames for transmission. Toffset is calculated as follows:~~

~~Toffset = 2 * average transmit duration of the series of concatenated frames~~

~~Toffset = 2 * Ttrans, 1.0 <= Toffset <= 10 seconds~~

~~where Ttrans = Average concatenated frame transmit duration~~

~~The average transmit duration of a series of concatenated frames is determined from the knowledge of the average length of the series of frames in bits divided by the effective on the air information transfer rate. The average length of the series of concatenated frames is computed based on the length of the last four series of concatenated frames transmitted by the local station. Toffset is bounded by 1.0 to throttle the channel by not allowing a station continuous access to the channel. The maximum of 10 seconds places an upper bound on the amount of time a station must wait between channel access attempts when long messages on a low rate channel are used. Toffset is to be updated after every transmission by the local station.~~

~~If the scheduler expires and there are no PDUs to transmit, the number of bits equal to one half the effective on the air information transfer rate (bps) will be entered into the record containing the last four transmissions. The value of Ttrans will default to 0.5 second. This will allow the Toffset parameter to default to 1 second during extended periods of inactivity.~~

C.4.4.4.1.3. Calculation of the scheduler random parameter. The parameter ~~schedint~~ SchedulerInterval depends on queue lengths and ~~average concatenated frame transmit duration~~ the number of net members transmitting data as follows:

~~schedint~~ SchedulerInterval = ~~Fsched * Ttrans, min <= schedint <= max~~[NADScaleTime * (NumActiveMembers/16)] * Fload
minimum=(settable range=0.1 to 3.0 sec)
maximum=(settable range=1.0 to 50 sec)

where:

~~Ttrans = Average concatenated frame transmit duration.~~NADScaleTime = adjustment factor (range=0.1 – 5.0 sec)

~~Fsched = Scheduling Factor. min = 3 seconds; max = 20 seconds.~~NumActiveMembers = number of net members actively transmitting data on the net. All net members are within two intranet hops of the node determining this value. Continuous calculation of this value shall be performed based on the number of known active data transmitters on the net. A station's own status is included in this count.

Neighbor agent timer – this timer (range=1 – 600 sec) is used to age out net member active status. When this timer expires and a data transmission has not been received from a net member, that net member is marked inactive. This timer should be greater than the topology update timer to assure net members are not aged out between topology updates in fragmented nets.

~~schedint~~ SchedulerInterval shall be recomputed after every transmission by the DTE.

C.4.4.4.1.4. Calculation of the ~~scheduling~~ Load factor (Fload). ~~The scheduling factor, Fsched, is based on three other factors: 1) the Partition Factor, Fpart, 2) the Topology Factor, Ftop, and 3) the Load Factor, Fload as follows:~~

~~Fsched = ((T*Ftop) (Fload))/((P*Fpart) + D)~~

~~such that min<=Fsched<=max~~

~~where:~~

	FH	SC
T =	1	2
P =	3	3
D =	7	7
min =	1	1
max =	20	20

~~and~~

~~FH = Frequency Hopping Mode~~
~~SC = Single Channel Mode~~
~~T = Topology Coefficient~~
~~P = Partition Coefficient~~
~~D = Damping Coefficient~~

~~Fload and Fsched are recomputed after every transmission attempt by the local station. Fpart and Ftop are computed after a topology update is generated or received. The T, P and D coefficients represent a compromise between high throughput and low delay, while promoting channel stability. The increase in coefficient T for single channel mode is compensating for the hidden terminal effect in multi-hop fixed frequency networks. The minimum value of 1 second has a throttling effect on the amount of time a station must wait between channel access attempts in a heavily congested, large network with multiple relayers.~~

~~C.4.4.4.1.4.1. Calculation of the Partition Factor. The Partition Factor is computed in a fully distributed manner by each node in the net. Partition Factor takes into account the one-hop connectivities experienced by each node in the network and is a measure of the connectivity between a node's neighbors. When a node's neighbors are strongly interconnected, i.e., the neighbors can hear each other, traffic will be routed directly between neighbors without a need for the node in question to relay the traffic. However, when a node's neighbors are weakly interconnected, for example the neighbors cannot hear each other, the node in question will see an increase in traffic due to relaying between neighbors.~~

~~The Partition Factor takes values between 1 for a strongly connected network and 7 for a weakly connected network. In the case of a higher partition factor the channel access scheduling interval is decreased, (the scheduling rate is increased), at the node doing the calculation, to meet the need for a higher percentage of the channel capacity to handle increased relay traffic. The Partition Factor for a non-relay node is set to 1 without executing the following algorithm.~~

~~Partition Factor is computed after a topology change is detected.~~

~~ALGORITHM: Calculation of the Partition Factor, Fpart, at node *j*.~~

~~1. Set the number of neighbors to zero, set the number of broken links to zero.~~

~~2. For each node *i* in the network, if there is a one hop link from *i* to *j* then:~~

~~2a. Increment the number of neighbors.~~

~~2b. For each node *k* in the network, if there is a one hop link from *k* to *j* and there is no one hop link from *i* to *k* then increment the number of broken links.~~

~~3. max links = # of neighbors * (# of neighbors -1)~~

~~4. If max links less than 1 then max links = 1.~~

~~5. Fpart = broken links * 6 / max links.~~

~~6. Fpart = Fpart + 1.~~

~~7. FINISH.~~

~~The constant 6 in algorithm step 5 establishes a set of 7 different levels of expected relaying. Nodes that are expected to do a significant amount of relaying (because their neighbors are not strongly connected) will receive the value 7 for Fpart, which shortens their scheduling interval per the algorithm in C.4.4.4.1.4. Nodes that are not expected to do a significant amount of relaying (because their neighbors are fully connected, thus reducing the amount of relaying required and providing a number of nodes to share relaying responsibilities) will receive the value 1. This lengthens their scheduling interval Nodes whose neighbors are neither strongly nor weakly connected are assigned a value between 1 and 7, exclusive, that depends on the degree of connectivity between the node's neighbors. This provides a total range of 7 values to bias a node's scheduler, depending upon the degree of relaying that a node is expected to do.~~

~~C.4.4.4.1.4.2. Calculation of the Topology Factor. The Topology Factor is computed in a fully distributed manner by each node in the net. This algorithm computes the ratio of the number of nodes that are one and two hops away from the node doing the computation, to the number of neighbors of the node doing the computation. For the Topology Factor Calculation the following applies: (1) a neighbor is defined as a node which is one hop away from the node doing the calculation, and (2) a neighbor node can also be categorized as a two-hop away node, from the node doing the computation, as long as there is a way of reaching that node in two hops (i.e., a neighbor node can simultaneously be listed, if it meets the criteria, as a two-hop away node). Since this is computed at each node in a fully distributed manner, this enables each node to evaluate its own situation in the network with respect to competition for the shared channel. A node with a high ratio of nodes within two hops to number of neighbors should use a longer scheduling interval due to the fact that neighbor nodes will have to handle the relaying traffic between the node under consideration and all other nodes in the net. Also, neighbor nodes will experience a higher ratio of receive collisions since the node in question and the nodes two hops~~

~~away are "hidden" from each other and thus cannot cooperate well in the channel access process.~~

~~Topology Factor takes values from 3 for a low ratio of nodes within two hops to neighbors, and 40 for a high ratio of nodes two hops to neighbors. In cases of higher Topology Factor values it is desirable to use a longer channel access scheduling interval to reduce the occurrence of channel access contention and collisions.~~

~~Topology Factor is computed after a topology change is detected.~~

~~ALGORITHM: Calculation of Topology Factor, Ftop, for node *j*~~

~~1. Set "number of neighbors" to zero.~~

~~2. Set "hearing within two hops" to zero.~~

~~3. For each node *i* in the net, if there is a one hop link from unit *i* to unit *j* then~~

~~3a. Increment the number of neighbors.~~

~~3b. Hearing within two hops = hearing within two hops + (number of neighbors of unit *i*) - 1. (don't count unit *j*).~~

~~4. If the number of neighbors is zero then set Ftop = 10. Go to step 10.~~

~~5. Hearing within 2 hops = hearing within 2 hops + # of neighbors.~~

~~6. Hearing within 2 hops = (hearing within 2 hops * 6)/4.~~

~~7. Ftop = hearing within two hops/number of neighbors.~~

~~8. If Ftop is less than 3 then Ftop = 3.~~

~~9. If Ftop is greater than 40 then Ftop = 40.~~

~~10. FINISH.~~

~~The constants 6 and 4 in algorithm step 6, and the constant 10 in algorithm step 4 are the default values which, when used in the Fsched equation (see C.4.4.4.1.4), restrict the channel access opportunities to a stable region offering good throughput and delay characteristics. The constants 6 and 4 in algorithm step 6 throttle the scheduler as the number of nodes increases, which preserves throughput. The values 6 and 4 give more range when using integer division than the values 3 and 2, respectively.~~

~~C.4.4.4.1.4.3. Calculation of the Load Factor.~~ The Load Factor is computed in a fully
distributed manner by every node in the net. In the transmission header, ~~one byte is~~ the six least significant bits of the second octet (as indicated in the Queue Precedence and Queue Length subfields of Figure 10 in section 5.3.1 of this document) are dedicated to transmitting the number of messages at the highest priority level remaining in the information frame queue; the remaining two bits are spare. The ~~four most significant bits (MSB)~~two least significant bits (LSB) indicate the level of the

highest priority message. The ~~three least~~four most significant bits (M~~L~~SB) indicate the number of frame concatenations~~'s~~ required to transmit all of the messages at the above priority level. The four ~~LSB~~ MSB are quantified as shown in Table C-1.

TABLE C-1. Calculation of the load factor.

Number of Concatenated Frames Required	Figure 10 Bit Pattern ~~(LSB is on the right)~~ bit 7 6 5 4
0.0	0 0 0 0
0.0 (exclusive) - 0.5 (inclusive)	0 0 0 1
0.5 (exclusive) - 1.0 (inclusive)	0 0 1 0
1.0 (exclusive) - 2.0 (inclusive)	0 0 1 1
2.0 (exclusive) - 3.0 (inclusive)	0 1 0 0
3.0 (exclusive) - 4.0 (inclusive)	0 1 0 1
4.0 (exclusive) - 5.0 (inclusive)	0 1 1 0
> 5.0	0 1 1 1

The Load Factor takes on values such that 1.0 < Fload < 18.0. The minimum value of 1.0 places an upper limit on the speed of the scheduler per the Fsched equation in C.4.4.4.1.4. The value of 18.0 provides a useful range for adaptation of the scheduler due to differing traffic loads, and is divisible by 2 and 3, resulting in integer ranges for the three different precedence values. Higher values of the Load Factor indicate that the node has shorter queues of equal or lesser priority. In cases of high load factor, it is desirable to increase the scheduling interval to give neighboring nodes with higher priority or longer queues of equal priority more opportunities to transmit. The Load Factor is calculated after every expiration of the scheduler, prior to calculation of the next expiration.

ALGORITHM: Calculation of the Load Factor, Fload.

1. Determine the number of unique neighboring node priority levels broadcast by all the nodes including this one. This data is taken from the last transmission received from each neighboring node.

2. Divide the interval 0.0 to 18.0 into equal segments, one per unique announced priority level. The first segment (0.0 to 9.0 for two levels) is allocated to the highest priority traffic. Define the Segment Offset as the lower bound of the chosen segment. For two precedence levels, the Segment Offset is 0.0 for the highest precedence and 9.0 for the Lowest Precedence. Define

the Segment width equal to 18.0/Number of precedence levels. For all three precedence levels, each precedence level has a segment width of 6.0

3. Each segment is subdivided into n unique levels where n is the number of unique quantified concatenated frame lengths reported by the neighboring nodes and the node doing the computation. In the case of only one length, all nodes use the midpoint of the segment. In the case of multiple lengths, these lengths are ordered from longest to shortest (1 -> n). In the following computation of Load Factor, a node would use a value of m determined by its position in that ordering. All nodes with the longest quantified length use the value 1, while those with the shortest use the value n for variable m in the following equation:

Load Factor = Segment offset + (Segment width*m)/(n+1)

where

Segment Offset is the Lower bound of the segment chosen by precedence level from step 2.

Segment Width is the maximum Load Factor (18) divided by the number of unique precedence levels

n is the number of unique quantified queue lengths.

m is this nodes positioning within the ordering of quantified queue lengths.

TABLE C-2. Calculation of the load factor -- example 1.

Node Number	Highest Precedence	Quantified Queue Length	Load Factor
1	Routine	0 0 0 1	12.0
2	Routine	0 0 0 1	12.0
3	Routine	0 0 1 1	6.0
4	Routine	0 0 1 1	6.0

All nodes compute the load factor in the following manner.

1. There is only 1 unique priority level (Routine).

2. The Segment is determined to encompass the whole range 0->18.

3. The Segment Offset is the lower bound (0).

4. The Segment Width is the entire range (18).

5. Two unique Quantified Queue Lengths are noted. The value of n is set to 2.

6. The unique Quantified Queue Lengths are ordered from longest to shortest (3,1).

7a. Nodes 1 and 2 note that their positioning in this sequence is 2 and set m to 2.

7b. Nodes 3 and 4 note that their positioning in this sequence is first and set m to 1.

8a. Nodes 1 and 2 compute their load factor from the equation.

Load_Factor = Segment Offset + (Segment Width*m)/(n+1)

= 0 + (18 * 2) / (2+1) = 12

8b. Likewise, Nodes 3 and 4 do the Load Factor computation.

Load_Factor = 0 + (18 * 1) / (2+1) = 6

TABLE C-3. Calculation of the load factor -- example 2.

Node Number	Highest Precedence	Quantified Queue Length	Load Factor
1	Routine	0 0 0 1	13.5
2	Routine	0 0 0 1	13.5
3	Urgent	0 0 1 0	6.0
4	Urgent	0 0 1 1	3.0

All nodes compute the load factor in the following manner.

1. There are two unique precedence levels (Urgent and Routine).

2. The load Factor Range is divided into two segments 0-9, 9-18. The segment 0-9 is reserved for Urgent, while the segment 9-18 is reserved for Routine.

3. The Segment Offset is the lower bound of the segment. The Segment Offset is 0 for Urgent and 9 for Routine.

4. The Segment Width for both precedence levels is the entire range (0->18) divided by the number of precedence levels. Segment Width = 18/2 = 9.

Nodes 1, 2 perform the following computations:

5. There is only one Quantified Queue Length. Thus, n is equal to 1 and since there is only 1 length both nodes use the first position in the sequence and set m to 1.

6. Load Factor = Segment Offset + (Segment Width*m)/(N+1)
   = 9 + (9 * 1) / (1+1) = 13.5

Nodes 3,4 perform the following computations.

7. The unique Quantified Queue Lengths are ordered from longest to shortest (3,2). There are two unique lengths which sets the value of n to 2.

8. Node 3 has a length of 2 which occupies position 2 in the ordering of step 7. Because is occupies position 2, the value of m is set to 2.

   Load_Factor = Segment Offset + (Segment Width*m)/(n+1)
   = 0 + (9 * 2) / (2+1) = 6

   Node 4 has length of 3 which occupies position 1 in the ordering of step 7. Node 4 sets it's value of m to 1.

   Load_Factor = Segment Offset + (Segment Width*m)/(N+1)
   = 0 + (9 * 1) / (2+1) = 3

C.4.4.4.1.5 100msec Immediate mode scheduling. ~~The average scheduling interval of the continuous scheduler is a factor in determining intranetwork end-to-end delay. In a lightly loaded network the average end-to-end delay will not be less than the average scheduling interval. In large nets this contributes to unnecessarily large end-to-end delay under conditions of low input load.~~

~~This situation is corrected by use of " immediate mode" scheduling under certain specific conditions.~~

~~The problem mentioned above is most obvious in large nets under conditions of light load. The Topology Factor incorporates network size in computing the scheduling interval increases as network size increases. However, in large nets under conditions of light input load channel utilization is low, yet end-to-end will be unnecessarily large due to the average scheduling interval of the continuous scheduler.~~

~~This situation is corrected using "~~In lightly loaded nets quicker access to the radio net medium is provided through the " 100msec immediate mode" scheduling all types except Type 1 ACK PDUs as follows:

~~a. If the message is Type 1 and requires a coupled acknowledgment, set Tc to 0.0 seconds and initiate an immediate channel access attempt. If the DCE is busy, implement the 1-persistent DTE-DC channel access protocol and transmit when the busy period ends. All stations receiving this transmission will suspend their Tc timers and observe the Type 1 timing for coupled acknowledgments.~~

a~~b~~. If the scheduler expires and the following conditions are met: (1)there are no concatenated frames awaiting transmission, (2) the RE-NAD voice factor is at its minimum value, and (3) all other members of the net reported transmission queue lengths of zero; set Immediate Mode ~~true~~to READY. Compute and start the next random interval of the continuous scheduler (Tc).

b~~c~~. When an information PDU arrives for transmission and Immediate Mode is ~~true~~READY, cancel the previuosly scheduled Tc ~~compute~~ and assign a scheduling interval as follows:

$$Tc = 100 \text{ msec}$$

c~~d~~. When this is done, Immediate Mode is ~~reset to false. The previously scheduled Tc is to be canceled~~set to ON. The 100 msec allows an opportunity for messages which have been segmented/fragmented/received to be piggy-backed into the same series of concatenated frames. This increases efficiency without imposing delay.

~~e.~~d. When the scheduler expires due either to the Tc scheduled as a result of the immediate mode operation or due to normal continuous operation,

~~MIL-STD-188-220B: 20 January 1998~~

and I-frame(s), S-frame(s), U-frame(s) or a frame concatenation are pending, perform concatenated frame processing as normally is done.

Compute and start the next random interval of the continuous scheduler (Tc) in the normal manner.

e~~f. The Tc interval timer set as a result of immediate mode operation is to be suspended and resumed for voice operation as is done for continuous mode operation.~~Any traffic, voice or data, detected during the immediate mode operation shall abort the 100msec Immediate Mode and set it to OFF. The scheduler is then computed in the normal manner.

C4.4.4.1.6Immediate mode scheduling. If the PDU is Type 1 (ACK or UnACK), the Tc is set to 0.0 seconds which permits net access to be controlled without a randomized scheduler influence.